

Study of Composite Materials using Zigzag Theory on Timoshenko Beams

M. Masó
E. Oñate
F. Zárte

Study of Composite Materials using Zigzag Theory on Timoshenko Beams

M. Masó
E. Oñate
F. Zárate



Publication CIMNE N°-397, June 2013

Register for free at <https://www.scipedia.com> to download the version without the watermark

International Center for Numerical Methods in Engineering
Gran Capitán s/n, 08034 Barcelona, Spain

Abstract

This *final studies work* target is to contribute two new beam theories to *MAT-fem* [6], an educational developed program by CIMNE [5]. Until now the *MAT-fem* Beams program only offered the Euler-Bernoulli and Timoshenko beam theories for homogeneous materials. With this work, the Timoshenko theory for composite materials and the refined zigzag theory are added.

MAT-fem Beams application works by discretizing beams into two noded elements. This work offers a beam theories opportunity of contrasting with up to four kinematic variables. Timoshenko theory for composite materials works with three kinematics variables per node and zigzag theory works with four variables. Finally, the accuracy of zigzag theory must be remarked in comparison to other classic beam theories.

Keywords

Register for free at <https://www.scipedia.com> to download the version without the watermark

Finite element method, *MAT-fem*, Timoshenko theory, Zigzag kinematics, Two-noded beam element, Composite material

Miguel Masó Sotomayor



Register for free at <https://www.scipedia.com> to download the version without the watermark

Contents

Abstract	i
----------	---

Contents	iv
----------	----

List of variables	v
-------------------	---

1 Introduction	1
----------------	---

1.1 Composite laminated plane beams	1
---	---

1.2 <i>MAT-fem</i> program	2
--------------------------------------	---

1.3 Organization	3
----------------------------	---

2 Beam theories	5
-----------------	---

2.1 Timoshenko composite laminated beams	5
--	---

2.1.1 Kinematics of a plane laminated beam	5
--	---

2.1.2 Stresses and resultant stresses	6
---	---

2.1.3 Generalized constitutive matrix	7
---	---

Layer defined constitutive matrix	7
---	---

Neutral axis	8
------------------------	---

Computation of the shear correction parameter	9
---	---

2.1.4 Thermal strains and initial stresses	11
--	----

2.1.5 Principle of virtual work	11
---	----

2.1.6 Two-noded composite laminated Timoshenko beam element	12
---	----

2.1.7 Shear locking	14
-------------------------------	----

2.2 Zigzag refined Timoshenko theory	15
--	----

2.2.1 General concepts of zigzag beam theory	15
--	----

2.2.2 Zigzag displacement field	16
---	----

2.2.3 Strain and stress fields	17
--	----

2.2.4 Computation of the zigzag function	18
--	----

2.2.5 Generalized constitutive matrix	19
---	----

Layer defined generalized constitutive matrix	20
---	----

2.2.6 Virtual work expression	21
---	----

Register for free at <https://www.scipedia.com> to download the version without the watermark

2.2.7	Two-noded LRZ beam element	22
2.2.8	Shear stresses integration	25
3	Numerical implementation	27
3.1	<i>MAT-fem.</i> The process	27
3.1.1	Start and read input file	27
	Input data file	27
	Zigzag-Timoshenko compatibility	33
3.1.2	Generalized constitutive matrix	33
	Timoshenko constitutive matrix	34
	Zigzag constitutive matrix	35
3.1.3	Elemental stiffness matrix and its assemble	38
3.1.4	External loads	40
3.1.5	Fixed displacements	41
3.1.6	Solution of the equation system	41
3.1.7	Reactions	42
3.1.8	Strains, stresses and resultant stresses	42
3.1.9	More about stresses evaluation	43
3.1.10	More about thickness distribution of strains and stresses	45
3.1.11	Writing for postprocessing	48
3.2	Graphical User Interface	49
3.2.1	Preprocess	49
	Details about the configuration files	53
3.2.2	Postprocess	55
4	Examples	57
4.1	Study of shear locking for the LRZ beam element	57
4.2	Convergence study	59
4.3	Modeling delamination	63
5	Conclusions	69
	Bibliography	71

Register for free at <https://www.scipedia.com> to download the version without the watermark

List of variables

Timoshenko theory files

Name	Size	Description
D_la	$n_{\text{layer}} \times 1$	Layer defined axial stiffness
D_lab	$n_{\text{layer}} \times 1$	Layer defined axial banding coupling stiffness
D_lb	$n_{\text{layer}} \times 1$	Layer defined bendig stiffness
D_ls	$n_{\text{layer}} \times 1$	Layer defined shear stiffness
D_mata	1×1	Axial stiffness
D_matab	1×1	Axial bending coupling stiffness
D_matb	1×1	Bending stiffness
D_mats	1×1	Shear stiffness
D_naxb	1×1	Bending stiffness in neutral axis
ElemFor	$1 \times n_{\text{node}}$	Elemental force vector
FreeNodes	$1 \times (n_{\text{ndof}} - \text{fix})$	List of the free DOF
K_axbn	$d_{\text{ofpe}} \times d_{\text{ofpe}}$	Axial bending coupling elemental stiffness matrix
K_axial	$d_{\text{ofpe}} \times d_{\text{ofpe}}$	Axial elemental stiffness matrix
K_bend	$d_{\text{ofpe}} \times d_{\text{ofpe}}$	Bending elemental stiffness matrix
K_elem	$d_{\text{ofpe}} \times d_{\text{ofpe}}$	Elemental stiffness matrix
K_shear	$d_{\text{ofpe}} \times d_{\text{ofpe}}$	Shear elemental stiffness matrix
S1	1×1	Auxiliary variable to compute Kz
S2	1×1	Auxiliary variable to compute Kz
S3	1×1	Auxiliary variable to compute Kz
StifMat	$n_{\text{ndof}} \times n_{\text{ndof}}$	Global stiffness matrix
StrNod	$n_{\text{pnod}} \times 6$	Array containing nodal stresses
blayr	$n_{\text{layer}} \times 1$	Layers width
condnum	$1 \times \text{fix}$	Auxiliary variable to make conditions compatible with zigzag
const	1×1	Auxiliary variable to compute the stiffness matrix

Register for free at <https://www.scipedia.com> to download the version without the watermark

Name	Size	Description
<code>coord_x</code>	1×2	Elemental x-coordinates
<code>coordinates</code>	$\text{npnod} \times 1$	Matrix containing the global coordinates
<code>denss</code>	$\text{nlayr} \times 1$	Layers density
<code>dnaxs</code>	1×1	Neutral axis vertical coordinate from middle section
<code>dofpe</code>	1×1	Degrees of freedom per element
<code>dofpn</code>	1×1	Degrees of freedom per node
<code>elements</code>	$\text{nelem} \times 2$	Connectivity matrix
<code>eqnum</code>	$1 \times \text{dofpe}$	Global equation number for all DOF inside an element
<code>f1</code>	1×1	1st DOF in the elemental force vector
<code>f2</code>	1×1	2nd DOF in the elemental force vector
<code>file_name</code>	1	Input file name
<code>fix</code>	$1 \times \text{fix}$	List of the constrained DOF
<code>fixnodes</code>	$\text{fix} \times 3$	Restricted nodes specification matrix
<code>force</code>	$\text{nndof} \times 1$	Global force vector
<code>forceDC</code>	$\text{nndof} \times 1$	Global force vector with the Dirichlet conditions contribution
<code>hlayr</code>	$\text{nlayr} \times 1$	Layers height
<code>hsect</code>	1×1	Beam depth
<code>ieqn</code>	1×1	Auxiliary variable to apply load and Dirichlet conditions
<code>ii</code>	1×1	Auxiliary variable to build global equation number
<code>itim</code>	1×1	Close the time counter
<code>kz</code>	1×1	Shear correction parameter
<code>lDispNod</code>	$\text{npnod} \times 1 \times (\text{nlayr}+1)$	Array containing the layer nodal displacements
<code>lResStrGP</code>	$\text{nelem} \times 5 \times \text{nlayr}$	Array containing the layer resultant stresses at the integration points
<code>lStrNod</code>	$\text{npnod} \times 3 \times (\text{nlayr}+2)$	Array containing the layer nodal stresses
<code>len</code>	1×1	Element lenght
<code>lnods</code>	1×2	Global node number for the element nodes
<code>mlayr</code>	$\text{nlayr} \times 1$	Vertical coordinate of the midpoint of the layers
<code>mmaxs</code>	3×1	Vertical coordinate of the midpoint of the layers from neutral axis
<code>nelem</code>	1×1	Number of elements
<code>nlayr</code>	1×1	Number of layers
<code>nndof</code>	1×1	Number of total DOF
<code>nnode</code>	1×1	Number of nodes per element
<code>npnod</code>	1×1	Number of total nodes
<code>pointload</code>	$\text{n} \times 3$	Punctual loads specification matrix

Register for free at <https://www.scipedia.com> to download the version without the watermark

CONTENTS

Name	Size	Description
poiss	$\text{nlayr} \times 1$	Layers Poisson value
reaction	$\text{nndof} \times 1$	Vector of global reactions
shear	$\text{nlayr} \times 1$	Layers Shear modulus
ttim	1×1	Time counter
u	$\text{nndof} \times 1$	Vector of nodal displacements
unload	$\text{nelem} \times 2$	Uniform loads specification matrix
weight	1×1	Section weight per unit length
young	$\text{nlayr} \times 1$	Layers Young modulus
zlayr	$(\text{nlayr}+1) \times 1$	Vertical coordinate of the layers limits
znaxs	$(\text{nlayr}+1) \times 1$	Vertical coordinate of the layers limits from neutral axis

Zigzag theory files

Name	Size	Description
B_matp	$3 \times \text{nnode}$	Transverse shear strain matrix
D_lpf	$3 \times 3 \times \text{nlayr}$	In-plane layer defined generalized constitutive matrix
D_ltf	$2 \times 2 \times 3 \times \text{nlayr}$	Transverse layer defined generalized constitutive matrix
D_matp	3×3	In-plane generalized constitutive matrix
D_matt	2×2	Transverse generalized constitutive matrix
D_s	1×1	Shear stiffness
ElemFor	$1 \times \text{nnode}$	Elemental force vector
FreeNodes	$1 \times (\text{nndof}-\text{fix})$	List of the free DOF
G	1×1	Equivalent shear modulus
K_elem	$\text{dofpe} \times \text{dofpe}$	Elemental stiffness matrix
K_p	$\text{dofpe} \times \text{dofpe}$	In-plane elemental stiffness matrix
K_psi	$\text{dofpe} \times \text{dofpe}$	Psi elemental stiffness matrix
K_s	$\text{dofpe} \times \text{dofpe}$	Shear elemental stiffness matrix
K_s_psi	$\text{dofpe} \times \text{dofpe}$	Shear Psi coupling elemental stiffness matrix
StifMat	$\text{nndof} \times \text{nndof}$	Global stiffness matrix
StrNod	$\text{npnod} \times 6$	Array containing nodal stresses
beta	$\text{nlayr} \times 1$	Zigzag function derivative
blayr	$\text{nlayr} \times 1$	Layers width
const	1×1	Auxiliary variable to compute the stiffness matrix
coor_x	1×2	Elemental x-coordinates
coor_z	150×1	Integration points to evaluate the shear stresses
coordinates	$\text{npnod} \times 1$	Matrix containing the global coordinates

Register for free at <https://www.scipedia.com> to download the version without the watermark

Name	Size	Description
denss	$nlayr \times 1$	Layers density
dofpe	1×1	Degrees of freedom per element
dofpn	1×1	Degrees of freedom per node
dz2	$nlayr \times 1$	Auxiliary variable to compute the genaralized constitutive matrix
dz3	$nlayr \times 1$	Auxiliary variable to compute the genaralized constitutive matrix
elements	$nelem \times 2$	Connectivity matrix
eqnum	$1 \times dofpe$	Global equation number for all DOF inside an element
f1	1×1	1st DOF in the elemental force vector
f2	1×1	2nd DOF in the elemental force vector
file_name	1	Input file name
fix	$1 \times fix$	List of the constrained DOF
fixnodes	$fix \times 3$	Restricted nodes specification matrix
force	$nndof \times 1$	Global force vector
forceDC	$nndof \times 1$	Global force vector with the Dirichlet conditions contribution
hlayr	$nlayr \times 1$	Layers height
hsect	1×1	Beam depth
ieqn	1×1	Auxiliary variable to apply load and Dirichlet conditions
ii	1×1	Auxiliary variable to build global equation number
itim	1×1	Close the time counter
lDspNod	$npnod \times 4 \times (nlayr+1)$	Array containing the layer nodal displacements
lDspStr	$nelem \times 9 \times nlayr$	Array containing the layer resultant stresses at the integration points
lStrNod	$npnod \times 3 \times (nlayr*2)$	Array containing the layer nodal stresses
len	1×1	Element lenght
lnods	1×2	Global node number for the element nodes
mlayr	$nlayr \times 1$	Vertical coordinate of the middpoint of the layers
nelem	1×1	Number of elements
nlayr	1×1	Number of layers
nndof	1×1	Number of total DOF
nnode	1×1	Number of nodes per element
npnod	1×1	Number of total nodes
phi	$(nlayr+1) \times 1$	Zigzag function
pointload	$n \times 3$	Punctual loads specification matrix
poiss	$nlayr \times 1$	Layers Poisson value
reaction	$nndof \times 1$	Vector of global reactions

Register for free at <https://www.scipedia.com> to download the version without the watermark

CONTENTS

Name	Size	Description
<code>shear</code>	$\text{nlayr} \times 1$	Layers Shear modulus
<code>shrStr</code>	$\text{npnod} \times 150 \times 2$	Array containing the integration of shear stresses
<code>ttim</code>	1×1	Time counter
<code>u</code>	$\text{nn dof} \times 1$	Vector of nodal displacements
<code>unload</code>	$\text{nelem} \times 2$	Uniform loads specification matrix
<code>weight</code>	1×1	Section weight per unit length
<code>young</code>	$\text{nlayr} \times 1$	Layers Young modulus
<code>zlayr</code>	$(\text{nlayr}+1) \times 1$	Vertical coordinate of the layers limits

Miguel Masó Sotomayor

Chapter 1

Introduction

1.1 Composite laminated plane beams

The advantages in strength and weight of composite materials versus traditional concrete and steel have led to an increased number of applications in engineering. The design of efficient and reliable composite structures requires improved computational methods that can accurately incorporate key mechanical effects [4].

Composite beams are typically formed by a piling of layers of composite material. The finite element analysis of the so called *composite laminated beams* has to account for the non-uniform distribution of the material properties along the beam thickness direction. Timoshenko beam theory is particularly suited to these problems as the heterogeneity of the material increases the importance of transverse shear deformation.

However the classical Euler-Bernoulli beam theory and the more advanced Timoshenko theory produce inadequate predictions when applied to relatively thick composite laminated beams with material layers that have highly different stiffness.

Improvements to the classical beam theories have been obtained by the so called equivalent layer (ESL) theories [3] that assume a priori the behavior of the displacement and/or stress through the laminate thickness. Despite being computationally efficient, ESL theories often produce inaccurate distributions for the stresses and strains across the thickness.

The need for composite laminated beam theories with better predictive capabilities has led to the development of the so-called *higher order* theories. In this theories

higher-order kinematic terms with respect to the beam depth are added to the expression for the axial displacement.

Accurate predictions of the correct shear and axial stresses for thick and highly heterogeneous composite laminated and sandwich beams can be obtained by using *layer-wise* theory. In this theory the thickness coordinate is split into a number of *analysis layers* that may or not coincide with the number of laminate plies. The kinematics are independently described within each layer and certain physical continuity requirements are enforced.

Discrete layer theories where the number of unknowns in the model does not depend on the number of layers in the laminate are called *zigzag* theories. In this class of discrete layer-wise theories a piecewise in-plane displacement function is superimposed over the displacement field over the thickness of the laminate.

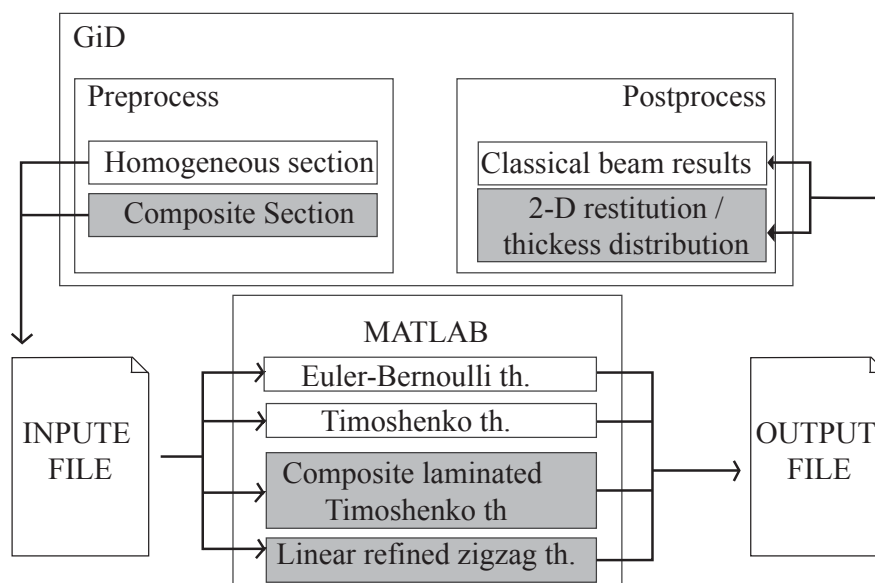
Oñate et al. [3] and [4] proposed a simple 2-noded beam element for composite laminated beams based on the refined zigzag theory. A standard linear displacement field is used to model the four variables of the linear refined zigzag (LRZ) element. Shear locking is avoided by using reduced integration on selected terms of the shear stiffness matrix.

1.2 *MAT-fem* program

MAT-fem has been produced under the close interaction of GiD [7] with MATLAB [9]. GiD allows manipulating geometries and discretizations while writing the input is required by MATLAB. The calculation program is executed in MATLAB without losing any of the MATLAB advantages. Finally GiD gathers the output data files for graphical visualization and interpretation.

This scheme allows understanding in detail the execution a Finite Elements program, following step by step each one of the code lines. At the same time is possible to run examples that by their dimensions would fall outside any program with educative aims.

MAT-fem has 2D elasticity applications and also heat transfer, beams, plates, shells, axisymmetric shells and sound transfer. This work focus in beams application and gives two new beam theories about composite materials. Figure 1.1 shows *MAT-fem* scheme, every item related to composite beams is gray shaded.

Figure 1.1: *MAT-fem* flow chart

1.3 Organization

The scheme organized of the subsequent chapters is as follows:

Beam theories concepts are presented in Chapter 2, followed by the basics of the finite element method formulation. Chapter 3 shows in detail the numerical implementation of the finite element method, and shear locking is discussed. Chapter 3 also explains preprocess and postprocess. Examples are included in Chapter 4 to test the reduced integration proposed in Chapter 3. The examples show convergence and capabilities of refined beam theories showed in Chapter 2. Finally, some conclusions are discussed in Chapter 5.

Miguel Masó Sotomayor

Chapter 2

Beam theories

2.1 Timoshenko composite laminated beams

2.1.1 Kinematics of a plane laminated beam

Let us consider a straight beam of length L and axis x linking the gravity centers G of all cross-sections with xz being a principal plane of inertia. The cross-section is formed by a piling of layers of composite material. Hence, in general the beam axis does not coincide with the neutral axis. The loads are vertical forces and bending moments contained in the xz plane as usual for plane beams. Bending on the plane yz will not be considered here.

Timoshenko hypothesis for the rotation of the normal to hold will be assumed. The axial and vertical displacements of a point A of the beam section are expressed as

$$u(x, z) = u_0(x) - z\theta(x) \quad ; \quad w(x, z) = w_0(x) \quad (2.1)$$

where $(\cdot)_0$ denotes the displacements of the beams axis

The axial and transverse shear strains are deduced from equations (2.1) as

$$\varepsilon_x = \frac{\partial u}{\partial x} = \frac{\partial u_0}{\partial x} - z \frac{\partial \theta}{\partial x} \quad (2.2a)$$

$$\gamma_{xz} = \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} = \frac{\partial w_0}{\partial x} - \theta \quad (2.2b)$$

Equation (2.2) can be written in matrix form as

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_x \\ \gamma_{xz} \end{Bmatrix} = \begin{bmatrix} 1 & -z & 0 \\ 0 & 0 & 1 \end{bmatrix} \left[\frac{\partial u_0}{\partial x}, \frac{\partial \theta}{\partial x}, \frac{\partial w_0}{\partial x} - \theta \right]^T = \mathbf{S} \hat{\boldsymbol{\varepsilon}} \quad (2.3a)$$

with

$$\mathbf{S} = \begin{bmatrix} 1 & -z & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \hat{\boldsymbol{\varepsilon}} = \left[\frac{\partial u_0}{\partial x}, \frac{\partial \theta}{\partial x}, \frac{\partial w_0}{\partial x} - \theta \right]^T \quad (2.3b)$$

where $\boldsymbol{\varepsilon}$ is the strain vector, $\hat{\boldsymbol{\varepsilon}}$ is the generalized strain vector containing the elongation of the beam axis $\left(\frac{\partial u_0}{\partial x}\right)$, the curvature $\left(\frac{\partial \theta}{\partial x}\right)$ and the transverse shear strain $\left(\frac{\partial w_0}{\partial x} - \theta\right)$ and \mathbf{S} is a strain-displacement transformation matrix depending on the thickness coordinate z .

2.1.2 Stresses and resultant stresses

The axial and shear stresses are expressed from equation (2.2) as

$$\sigma_x = E \varepsilon_x = E \left(\frac{\partial u_0}{\partial x} - z \frac{\partial \theta}{\partial x} \right) \quad (2.4a)$$

$$\tau_{xz} = G \gamma_{xz} = G \left(\frac{\partial w_0}{\partial x} - \theta \right) \quad (2.4b)$$

where $E = E(x, z)$ and $G = G(x, z)$ are the longitudinal Young modulus and the shear modulus of the beam composite material.

Equation (2.4) can be written in matrix form using equation (2.3) as

$$\boldsymbol{\sigma} = \begin{Bmatrix} \sigma_x \\ \tau_{xz} \end{Bmatrix} = \begin{bmatrix} E & 0 \\ 0 & G \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \gamma_{xz} \end{Bmatrix} = \mathbf{D} \boldsymbol{\varepsilon} = \mathbf{D} \mathbf{S} \hat{\boldsymbol{\varepsilon}} \quad (2.5)$$

where \mathbf{D} is the standard constitutive matrix relating stresses and strains at a point in the transverse cross section.

The axial force N , the bending moment M and the shear force Q in a beam section are obtained as

$$\hat{\boldsymbol{\sigma}} = \begin{Bmatrix} N \\ M \\ Q \end{Bmatrix} = \iint_A \begin{Bmatrix} \sigma_x \\ -z \sigma_x \\ \tau_{xz} \end{Bmatrix} dA = \iint_A \mathbf{S}^T \boldsymbol{\sigma} dA \quad (2.6)$$

where $\hat{\boldsymbol{\sigma}}$ is the resultant stress vector and A is the area of the cross-section.

2.1.3 Generalized constitutive matrix

Substituting equation 2.5 into 2.6 gives

$$\hat{\sigma} = \left(\iint_A \mathbf{S}^T \mathbf{D} \mathbf{S} dA \right) \hat{\varepsilon} = \hat{\mathbf{D}} \hat{\varepsilon} \quad (2.7)$$

where $\hat{\varepsilon}$ is the generalized strain vector defined in equation (2.3b) and $\hat{\mathbf{D}}$ is the generalized constitutive matrix. The terms of $\hat{\mathbf{D}}$ are computed as

$$\hat{\mathbf{D}} = \iint_A \mathbf{S}^T \mathbf{D} \mathbf{S} dA = \begin{bmatrix} \hat{D}_a & \hat{D}_{ab} & 0 \\ \hat{D}_{ab} & \hat{D}_b & 0 \\ 0 & 0 & \hat{D}_s \end{bmatrix} \quad (2.8a)$$

with

$$\begin{aligned} \hat{D}_a &= \iint_A E(x, z) dA \quad ; \quad \hat{D}_{ab} = - \iint_A E(x, z) z dA \\ \hat{D}_b &= \iint_A E(x, z) z^2 dA \quad ; \quad \hat{D}_s = k_z \hat{G} \quad \text{with } \hat{G} = \iint_A G(x, z) dz \end{aligned} \quad (2.8b)$$

where \hat{D}_a is the axial stiffness, \hat{D}_b is the bending stiffness, \hat{D}_{ab} is the coupling axial-bending stiffness, \hat{D}_s is the shear stiffness and k_z is the shear correction parameter for bending around the y axis. The computation of k_z is explained in the next section.

Layer defined constitutive matrix

From equations (2.8b) we can define the layer defined generalized constitutive matrix as

$$\begin{aligned} \hat{D}_a &= \sum_{k=1}^{n_l} [\hat{D}_a]^k & \hat{D}_b &= \sum_{k=1}^{n_l} [\hat{D}_b]^k \\ \hat{D}_{ab} &= \sum_{k=1}^{n_l} [\hat{D}_{ab}]^k & \hat{D}_s &= \sum_{k=1}^{n_l} [\hat{D}_s]^k \end{aligned} \quad (2.9)$$

This formulation results usefully when computing resultant stresses. For a laminated beam with n_l layers of isotropic material with modulae E^k , G^k , thickness

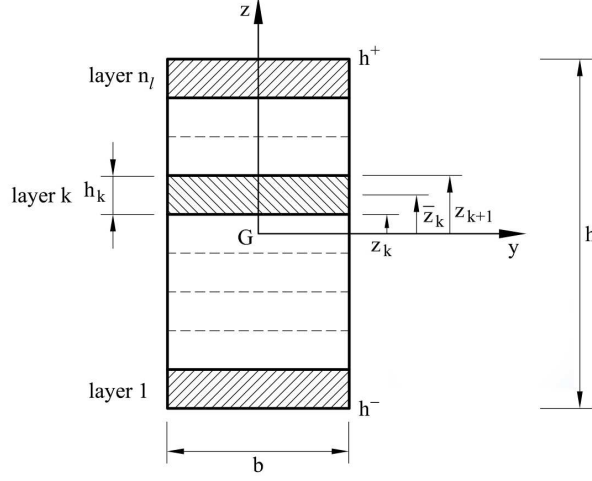


Figure 2.1: Rectangular laminated beam. Coordinates and nomenclature to integrate the material properties

h_k and width b_k we have

$$\begin{aligned}
 [\hat{D}_a]^k &= b(z_{k+1} - z_k)E^k = bh_k E^k \\
 [\hat{D}_{ab}]^k &= -\frac{b}{2}(z_{k+1}^2 - z_k^2)E^k = -bh_k \bar{z}_k E^k \\
 [\hat{D}_b]^k &= \frac{b}{3}(z_{k+1}^3 - z_k^3)E^k \\
 [\hat{D}_s] &= k_z b(z_{k+1} - z_k)G^k = k_z bh_k G^k
 \end{aligned} \tag{2.10}$$

where \bar{z}_k is the vertical coordinate of the midpoint of the k th layer. Figure 2.1 shows an example of a rectangular laminated beam.

Neutral axis

The position of the *neutral axis* for an arbitrary composite laminated section can be found as follows. Let us define the relative vertical coordinate $z' = z - d$ where d is the vertical distance between the beam axis x and the neutral axis. If the x axis is placed at point O defining the neutral axis (figure 2.2), then

$$\hat{D}_{ab} = - \iint_A E z' dA = - \iint_A E (z - d) dA = 0 \tag{2.11}$$

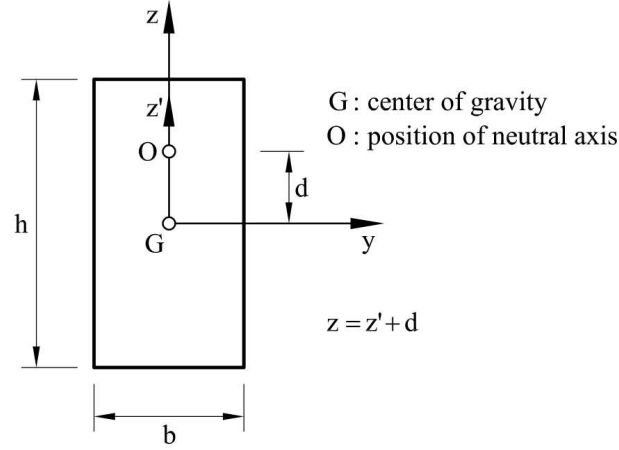


Figure 2.2: Position of the neutral axis

From equations (2.11) and (2.8) we can obtain

$$d = -\frac{\hat{D}_{ab}}{\hat{D}_a} \quad (2.12)$$

Computation of the shear correction parameter

In this section is used the expression for the shear correction parameter given in [3]

$$k_z = \frac{Q^2}{\hat{G}} \left[\iint_A \frac{\tau_{xz}^2}{G(z)} \right]^{-1} dA \quad (2.13)$$

Inverting equation (2.9) and substituting it into (2.4) gives σ_x at each layer in terms of N and M by

$$\sigma_x = \frac{E}{\hat{D}} \left[\hat{D}_b N - \hat{D}_{ab} M - z(-\hat{D}_{ab} N + \hat{D}_a M) \right] \quad (2.14)$$

Equation (2.4) shows that the shear stress τ_{xz} is constant across the beam depth. The “correct” distribution of τ_{xz} which satisfies the equilibrium equations of elasticity can be computed “a posteriori” once the displacements have been obtained. From the equilibrium equation along the x direction

$$\frac{\partial \tau_{xz}}{\partial z} + \frac{\partial \sigma_x}{\partial x} = 0 \quad \rightarrow \quad \tau_{xz}(z) = - \int_{h^-}^z \frac{\partial \sigma_x}{\partial x} dz \quad (2.15)$$

Miguel Masó Sotomayor

Substituting equation 2.14 into equation 2.15 and accepting that $\frac{\partial N}{\partial x} = 0$ and using $\frac{\partial M}{\partial x} = -Q$ gives

$$\tau_{xz}(z) = \frac{-Q}{\hat{D}} F(z) \quad (2.16)$$

with

$$F(z) = \hat{D}_a S(z) + \hat{D}_{ab} \int_{h^-}^z E(z) dz \quad , \quad S(z) = \int_{h^-}^z z E(z) dz \quad (2.17)$$

If x is the neutral axis, then $\hat{D}_{ab} = 0$, and, hence, $\hat{D} = \hat{D}_a \hat{D}_b$ and

$$\tau_{xz}(z) = \frac{-Q}{\hat{D}_b} S(z) \quad (2.18)$$

Substituting equation (2.16) into (2.13) gives

$$k_z = \frac{\hat{D}^2}{\hat{G}} \left[\iint_A \frac{F^2(z)}{G(z)} dA \right]^{-1} \quad (2.19)$$

If x is the neutral axis, then $F = \hat{D}_a S(z)$, $\hat{D} = \hat{D}_a \hat{D}_b$ and

$$k_z = \frac{\hat{D}_b^2}{\hat{G}} \left[\iint_A \frac{S^2(z)}{G(z)} dA \right]^{-1} \quad (2.20)$$

Integrating equation (2.20) in a rectangular composite laminated section we obtain the next expression

$$\begin{aligned} \iint_A \frac{S^2(z)}{G(z)} dA &= \sum_{j=1}^{n_l} \frac{b}{G_i} \iint_{z_j}^{z_{j+1}} S^2(z) dA = \\ &= b \sum_{k=1}^{n_l} \frac{1}{4G^k} \left(\sum_{l=1}^{k-1} E^l \left((z^{l+1})^2 - (z^l)^2 \right) - E^k (z^k)^2 \right)^2 (z^{k+1} - z^k) \\ &+ b \sum_{k=1}^{n_l} \frac{1}{6G^k} \left(\sum_{l=1}^{k_1} E_l \left((z^{l+1})^2 - (z^l)^2 \right) - E^k (z^k)^2 \right) E^k \left((z^{k+1})^3 - (z^k)^3 \right) \\ &+ b \sum_{k=1}^{n_l} \frac{1}{20} (E^k)^2 \left((z^{k+1})^5 - (z^k)^5 \right) \quad (2.21) \end{aligned}$$

2.1.4 Thermal strains and initial stresses

An initial axial strain due to thermal effects (ε_x^o) and initial stresses (σ_x^o, τ_{xz}^o) can easily be accounted for in the present formulation. The strain-stress relationship of Eq.(3.8) is modified as

$$\sigma_x = E(\varepsilon_x - \varepsilon_x^o) + \sigma_x^o \quad ; \quad \tau_{xz} = G\gamma_{xz} + \tau_{xz}^o \quad (2.22)$$

where $\varepsilon_x^o = \alpha\Delta T$, α being the thermal expansion coefficient and ΔT the temperature increment. Recall that the initial tangential stresses due to a thermal expansion are zero (Section 4.2.4 of [On4]).

The relationship between resultant forces and generalized strains (equation (2.4)) is modified as

$$\hat{\boldsymbol{\sigma}} = \hat{\mathbf{D}}\hat{\boldsymbol{\varepsilon}} + \hat{\boldsymbol{\sigma}}^o \quad (2.23a)$$

where $\hat{\boldsymbol{\sigma}}^o$ is the initial resultant stress vector given by

$$\hat{\boldsymbol{\sigma}}^o = [N^o, M^o, Q^o]^T \quad (2.23b)$$

with

$$N^o = \iint_A [-E\varepsilon_x^o + \sigma_x^o] dA \quad , \quad M^o = \iint_A [E\varepsilon_x^o - \sigma_x^o] z dA \quad , \quad Q^o = \iint_A \tau_{xz}^o dA \quad (2.23c)$$

2.1.5 Principle of virtual work

We study the PVW for distributed loads \mathbf{t} only. Other load types(i.e. point loads) can be easily taken into account as explained in Zienkiewicz [1] and Oñate [2]. The expression of the PVW is

$$\iiint_V \delta \boldsymbol{\varepsilon}^T \boldsymbol{\sigma} dV = \int_L \delta \mathbf{u}^T \mathbf{t} dx \quad (2.24)$$

where $\delta \mathbf{u} = [\delta u_0, \delta w_0, \delta \theta]^T$ is the virtual displacement vector, $\delta \boldsymbol{\varepsilon}$ and $\boldsymbol{\sigma}$ are the virtual strain vector and the stress vector, respectively, and $\mathbf{t} = [f_x, f_z, m]^T$ is the vector of external forces acting over the beam axis due to distributed axial and vertical loads f_x and f_z , respectively and a distributed moment m . The integral in the l.h.s. of equation (2.24) represents the internal virtual work.

Making use of equations (2.3) and (2.5), equation (2.24) can be written as

$$\iiint_V \delta \boldsymbol{\varepsilon}^T \hat{\boldsymbol{\sigma}} dV = \int_l \delta \hat{\boldsymbol{\varepsilon}}^T \left[\iint_A \mathbf{S}^T \mathbf{D} \mathbf{S} dA \right] \hat{\boldsymbol{\varepsilon}} dx = \int_L \delta \hat{\boldsymbol{\varepsilon}}^T \hat{\mathbf{D}} \hat{\boldsymbol{\varepsilon}} dx = \int_L \delta \hat{\boldsymbol{\varepsilon}}^T \hat{\boldsymbol{\sigma}} dx \quad (2.25)$$

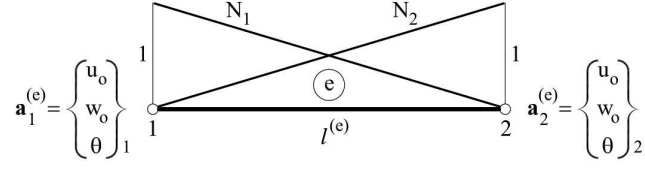


Figure 2.3: Two-noded composite laminated Timoshenko beam element

The PVW can therefore be expressed in terms of integrals along the beam axis as

$$\int_L \delta \hat{\boldsymbol{\varepsilon}}^T \hat{\boldsymbol{\sigma}} dx = \int_L \delta \mathbf{u}^T \mathbf{t} dx \quad (2.26)$$

all the derivatives appearing in the PVW are of first order. This allows us using C^0 continuous interpolations for the axial displacement u_0 , the vertical deflection w_0 and the rotation θ .

2.1.6 Two-noded composite laminated Timoshenko beam element

The beam is discretized into 2-noded elements of length $l^{(e)}$. A standard linear approximation is chosen for u_0 , w_0 and θ as (figure 2.3)

$$\mathbf{u} = \begin{Bmatrix} u_0 \\ w_0 \\ \theta \end{Bmatrix} = \sum_{i=1}^2 N_i(\xi) \mathbf{a}_i^{(e)} \quad \text{with} \quad \mathbf{a}_i^{(e)} = \begin{Bmatrix} u_0 \\ w_0 \\ \theta \end{Bmatrix}_i \quad (2.27)$$

where $(\cdot)_i$ denotes nodal values.

Substituting the approximation (2.27) into the generalized strain vector of equation (2.3a) gives

$$\hat{\boldsymbol{\varepsilon}} = \begin{Bmatrix} \frac{\partial u_0}{\partial x} \\ \frac{\partial \theta}{\partial x} \\ \frac{\partial w_0}{\partial x} - \theta \end{Bmatrix} = \sum_{i=1}^2 \mathbf{B}_i \mathbf{a}_i^{(e)} = \mathbf{B} \mathbf{a}^{(e)} \quad (2.28)$$

with

$$\mathbf{a}^{(e)} = \begin{Bmatrix} \mathbf{a}_1^{(e)} \\ \mathbf{a}_2^{(e)} \end{Bmatrix} \quad \text{and} \quad \mathbf{B}_i = \begin{Bmatrix} \mathbf{B}_{a_i} \\ \dots \\ \mathbf{B}_{b_i} \\ \dots \\ \mathbf{B}_{s_i} \end{Bmatrix} = \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 & 0 \\ \dots & \dots & \dots \\ 0 & 0 & \frac{\partial N_i}{\partial x} \\ \dots & \dots & \dots \\ 0 & \frac{\partial N_i}{\partial x} & -N_i \end{bmatrix} \quad (2.29)$$

where \mathbf{B}_{m_i} , \mathbf{B}_{b_i} and \mathbf{B}_{s_i} are the generalized strain matrices corresponding to axial, bending and transverse shear deformation effects.

Substituting the constitutive relationship (2.23) into the PVW (equation (2.26)) and using equations (2.27) and (2.28) leads to the system of equations $\mathbf{K}\mathbf{a} = \mathbf{f}$ where the stiffness matrix and the equivalent nodal force vector are assembled from the element contributions given by

$$\begin{aligned} \mathbf{K}_{ij}^{(e)} &= \int_{l^{(e)}} \mathbf{B}_i^T \hat{\mathbf{D}} \mathbf{B}_j dx \\ \mathbf{f}_i^{(e)} &= \begin{Bmatrix} f_{x_i} \\ f_{z_i} \\ m_i \end{Bmatrix} = \int_{l^{(e)}} N_i^{(e)} \mathbf{t} dx - \int_{l^{(e)}} \mathbf{B}_i^T \hat{\boldsymbol{\sigma}}^o dx \quad i, j = 1, 2 \end{aligned} \quad (2.30)$$

The second integral in the expression of $\mathbf{f}_i^{(e)}$ accounts for the effect of the initial (thermal) strain and the initial stresses.

The element stiffness matrix can be written using the components of \mathbf{B}_i and $\hat{\mathbf{D}}$ as

$$\mathbf{K}_{ij}^{(e)} = \mathbf{K}_{a_{ij}}^{(e)} + \mathbf{K}_{b_{ij}}^{(e)} + \mathbf{K}_{s_{ij}}^{(e)} + \mathbf{K}_{ab_{ij}}^{(e)} + [\mathbf{K}_{ab_{ij}}^{(e)}]^T \quad (2.31a)$$

where

$$\mathbf{K}_{r_{ij}}^{(e)} = \int_{l^{(e)}} \mathbf{B}_{r_i}^T \hat{D}_r \mathbf{B}_{r_j} dx \quad r = a, b, s \quad (2.31b)$$

and

$$\mathbf{K}_{ab_{ij}}^{(e)} = \int_{l^{(e)}} \mathbf{B}_{a_i}^T \hat{D}_{ab} \mathbf{B}_{b_j} dx \quad (2.31c)$$

In the above expressions indexes a, b, s and ab denote respectively the contribution of the axial, bending, shear and coupling axial-bending terms to the element stiffness matrix.

Finally, the explicit element stiffness matrix can be written as:

$$\mathbf{K}_a^{(e)} = \frac{\hat{D}_a}{l^{(e)}} \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.32a)$$

$$\mathbf{K}_b^{(e)} = \frac{\hat{D}_b}{l^{(e)}} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix} \quad (2.32b)$$

$$\mathbf{K}_{ab}^{(e)} = \frac{\hat{D}_{ab}}{l^{(e)}} \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.32c)$$

$$\mathbf{K}_s^{(e)} = \frac{\hat{D}_s}{l^{(e)}} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \frac{l^{(e)}}{2} & 0 & -1 & \frac{l^{(e)}}{2} \\ 0 & \frac{l^{(e)}}{2} & \frac{l^{(e)^2}}{3} & 0 & -\frac{l^{(e)}}{2} & \frac{l^{(e)^2}}{6} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & -\frac{l^{(e)}}{2} & 0 & 1 & -\frac{l^{(e)}}{2} \\ 0 & \frac{l^{(e)}}{2} & \frac{l^{(e)^2}}{6} & 0 & -\frac{l^{(e)}}{2} & \frac{l^{(e)^2}}{3} \end{bmatrix} \quad (2.32d)$$

2.1.7 Shear locking

The relative value of the shear stiffness terms versus the bending terms affects the finite element solution for the Timoshenko beam problem is explained in E. Oñate [2] and [3]. For thick beams, the shear terms dominate the bending ones in the stiffness matrix and this leads to unrealistically stiff results (locking). The relative influence of the shear terms over the bending terms can be quantified by the parameter β of equation (2.33):

$$\beta = \frac{12\hat{D}_b}{L^2\hat{D}_s} \quad (2.33)$$

A small value of β indicates that the influence of transverse shear deformation is negligible in the solution. Parameter β depends on the geometrical and mechanical properties of the section. For a rectangular beam of length L , depth h and homogeneous isotropic material, $\beta = E(k_z G \lambda^2)^{-1}$ with $\lambda = L/h$ being the beam slenderness ratio.

For a relatively “thick” isotropic beam ($\lambda = 4$), the ratio $\frac{E}{k_z G} \simeq 2$ and $\beta = 0.125$.

It is interesting that for a slender composite beam with $\lambda = 20$ and $\frac{E}{k_z G} = 50$ the value of β is also 0.125. The influence of transverse shear deformation is the same for a thick isotropic beam and a slender composite beam, both leading to a small value of β . This justifies using Timoshenko theory for composite laminated beams.

Shear locking appearing for small values of β can be eliminated by any of the methods explained in the previous chapter. For the 2-noded composite Timoshenko beam element, the simplest procedure is to evaluate all integrals in the stiffness matrix *using a single Gauss integration point*. Element shear stiffness matrix becomes as follows:

$$\mathbf{K}_s^{(e)} = \frac{\hat{D}_s}{l^{(e)}} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \frac{l^{(e)}}{2} & 0 & -1 & \frac{l^{(e)}}{2} \\ 0 & \frac{l^{(e)}}{2} & \frac{l^{(e)^2}}{4} & 0 & -\frac{l^{(e)}}{2} & \frac{l^{(e)^2}}{4} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & -\frac{l^{(e)}}{2} & 0 & 1 & -\frac{l^{(e)}}{2} \\ 0 & \frac{l^{(e)}}{2} & \frac{l^{(e)^2}}{4} & 0 & -\frac{l^{(e)}}{2} & \frac{l^{(e)^2}}{4} \end{bmatrix} \quad (2.34)$$

2.2 Zigzag refined Timoshenko theory

2.2.1 General concepts of zigzag beam theory

The displacement field in layer-wise theory is written as a linear combination of some function as

$$u_i(x, z) = u_i^0(x) + \sum_{k=1}^{N_i} u_i^k(x) \phi_k(z) \quad (2.35)$$

where N_i is the number of analysis layers taken, $u_i^k(x, y)$ are the displacements at each layer interface k and ϕ_j are known functions of the thickness coordinate

z . The ϕ_k functions are piecewise and continuous within each layer. Due to the local definition of $\phi_j(z)$, the displacements are continuous across the thickness but their derivatives with respect to z are not. So the transverse shear strains are discontinuous at the interfaces and the transverse shear stress can be enforced to be continuous for the case of layers with different mechanical properties.

The zigzag theories assume a zigzag pattern for the axial displacements and enforce continuity of the transverse shear stresses across the entire laminate depth. The number of kinematic variables in zigzag theories is *independent of the number of layers*. The kinematic field in zigzag beam theories is written as

$$u^k(x, z) = u_0(x) - z\theta(x) + \bar{u}^k(x, z) \quad ; \quad w(x, z) = w_0(x) \quad (2.36a)$$

where

$$\bar{u}^k = \phi^k(z)\Psi(x) \quad (2.36b)$$

is the *zigzag displacement function*.

Function $\phi^k(z)$ denotes a *piecewise linear zigzag function* and $\Psi(x)$ is a primary kinematic variable that defines the amplitude of the zigzag function along the beam.

2.2.2 Zigzag displacement field

The key attributes of the RZT are: first, *the zigzag function vanishes at the top and bottom surfaces of the beam section* and does not require full shear-stress continuity across the laminated-beam depth. Second, all boundary conditions can be modelled adequately. And third, C^0 continuity is only required for the FEM approximation of the kinematic variables.

Within each layer the zigzag function is expressed as

$$\phi^k = \frac{1}{2}(1 - \zeta)\bar{\phi}^{k-1} + \frac{1}{2}(1 + \zeta)\bar{\phi}_k^k = \frac{\bar{\phi}^k + \bar{\phi}^{k-1}}{2} + \frac{\bar{\phi}_k^k - \bar{\phi}^{k-1}}{2}\zeta^k \quad (2.37)$$

where $\bar{\phi}^k$ and $\bar{\phi}^{k-1}$ are the zigzag functions of the k and $k-1$ interface, respectively with $\bar{\phi}^0 = \bar{\phi}^n = 0$ and $\zeta^k = \frac{2(z - z^{k-1})}{h^k} - 1$ (figure 2.4a).

Note that the zigzag displacement \bar{u}^k (equation) also vanishes at the top and bottom layers (figure 2.4b).

The form of ϕ^k of equation (2.37) yields a constant distribution of its gradient within each layer β^k

$$\beta^k = \frac{\partial \phi^k}{\partial z} = \frac{\bar{\phi}^k - \bar{\phi}^{k-1}}{h^k} \quad (2.38a)$$

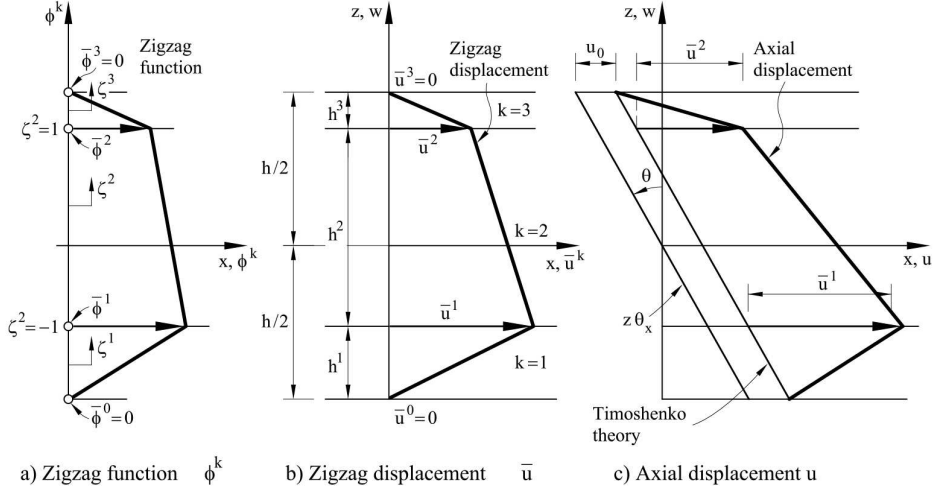


Figure 2.4: Refined zigzag theory. Zigzag and displacement fields

From equation (2.38) and the conditions $\bar{\phi}^0 = \bar{\phi}^N = 0$ we deduce

$$\iint_A \beta^k dA = 0 \quad (2.38b)$$

The β^k parameter is useful for computing the zigzag function.

2.2.3 Strain and stress fields

The strain-displacement relations are derived from equations (2.2) and (2.36a) as

$$\varepsilon_x^k = \frac{\partial u_0}{\partial x} - z \frac{\partial \theta}{\partial x} + \phi^k \frac{\partial \Psi}{\partial x} = [1, -z, \phi^k] \begin{Bmatrix} \frac{\partial u_0}{\partial x} \\ \frac{\partial \theta}{\partial x} \\ \frac{\partial \Psi}{\partial x} \end{Bmatrix} = \mathbf{S}_p \hat{\varepsilon}_p \quad (2.39a)$$

$$\gamma_{xz}^k = \gamma + \beta^k \Psi = [1, \beta^k] \begin{Bmatrix} \gamma \\ \Psi \end{Bmatrix} = \mathbf{S}_t^k \hat{\varepsilon}_t \quad (2.39b)$$

with

$$\begin{aligned} \mathbf{S}_p &= [1, -z, \phi^k] \quad , \quad \hat{\varepsilon}_p = \left[\frac{\partial u_0}{\partial x}, \frac{\partial \theta}{\partial x}, \frac{\partial \Psi}{\partial x} \right]^T \\ \mathbf{S}_t^k &= [1, \beta^k] \quad , \quad \hat{\varepsilon}_t = [\gamma, \Psi]^T \end{aligned} \quad (2.39c)$$

where $\hat{\boldsymbol{\varepsilon}}_p$ and $\hat{\boldsymbol{\varepsilon}}_t$ are the generalized in-plane (axial-bending) and transverse shear strain vectors, respectively.

In equation (2.39b), $\gamma = \frac{\partial w_0}{\partial x} - \theta$. Integrating equation (2.39b) over the cross section and using equation (2.38) and the fact that Ψ is independent of z yields

$$\gamma = \frac{1}{A} \iint_A \gamma_{xz}^k dA \quad (2.40)$$

i.e. γ represents the average transverse shear strain of the cross section. Stress-strain relations for the k th layer have the next form (equation 2.4)

$$\sigma_x^k = E^k \varepsilon_x^k = E^k \mathbf{S}_p^k \hat{\boldsymbol{\varepsilon}}_p \quad (2.41a)$$

$$\tau_{xz}^k = G^k \gamma_{xz}^k = G^k \mathbf{S}_t^k \hat{\boldsymbol{\varepsilon}}_t \quad (2.41b)$$

where E^k and G^k are the axial and shear moduli for the k th layer.

2.2.4 Computation of the zigzag function

The shear strain-shear stress relationship of equation (2.39b), is written as

$$\tau_{xz}^k = G^k \eta + G^k (1 + \beta^k) \Psi \quad (2.42)$$

where $\eta = \gamma - \Psi$ is a difference function.

Clearly the distribution of τ_{xz}^k within each layer is constant, as η is independent of the zigzag function and β^k is constant (equation (2.38)).

The distribution of τ_{xz}^k is enforced to be independent of the zigzag function. This can be achieved by constraining the term multiplying Ψ in equation (2.42) to be constant, i.e.

$$G^k (1 + \beta^k) = G^{k+1} (1 + \beta^{k+1}) = G, \quad \text{constant} \quad (2.43)$$

This is equivalent to enforcing the interfacial continuity of the second term in the r.h.s. of equation (2.42).

From equation (2.43) we deduce

$$\beta^k = \frac{G}{G^k} - 1 \quad (2.44)$$

Substituting β^k in the integral of equation (2.38) gives

$$G = \left[\frac{1}{A} \iint_A \frac{dA}{G^k} \right]^{-1} = \left[\frac{1}{h} \sum_{k=1}^{n_l} \frac{h^k}{G^k} \right]^{-1} \quad (2.45)$$

which is the *equivalent shear modulus* for the laminate.

Substituting equation (2.38a) into equation (2.38b) gives the following recursion relation for the zigzag function values at the layer interfaces

$$\bar{\phi}_k = \sum_{i=1}^k h^i \beta^i \quad \text{with} \quad u^0 = u^{n_l} = 0 \quad (2.46)$$

with β^i given by Eq.(3.58).

Introducing Eq.(3.60) into (3.51) gives the expression for the zigzag function as

$$\phi^k = \frac{h^k \beta^k}{2} (\zeta^k - 1) + \sum_{i=1}^k h^i \beta^i \quad (2.47)$$

This theory does not enforce the continuity of the transverse shear stresses across the section. For homogeneous material $G^k = G$ and $\beta^k = 0$. The zigzag function ϕ^k vanishes and we recover the kinematic and constitutive expressions of the standard Timoshenko composite laminated beam theory.

Function Ψ can be interpreted as a weighted-average shear strain angle [3]. *The value of Ψ should be prescribed to zero at a clamped edge and left unprescribed at a free edge.*

2.2.5 Generalized constitutive matrix

The resultant stresses are defined as

$$\hat{\sigma}_p = \begin{Bmatrix} N \\ M \\ M_\phi \end{Bmatrix} = \iint_A [\mathbf{S}_p^k]^T \sigma_x^k dA = \left(\iint_A [\mathbf{S}_p^k]^T \mathbf{S}_p^k E^k dA \right) \hat{\epsilon}_p = \hat{\mathbf{D}}_p \hat{\epsilon}_p \quad (2.48)$$

$$\hat{\sigma}_t = \begin{Bmatrix} Q \\ Q_\phi \end{Bmatrix} = \iint_A [\mathbf{S}_t^k]^T \tau_{xz}^k dA = \left(\iint_A [\mathbf{S}_t^k]^T \mathbf{S}_t^k G^k dA \right) \hat{\epsilon}_t = \hat{\mathbf{D}}_t \hat{\epsilon}_t \quad (2.49)$$

In vectors $\hat{\sigma}_p$ and $\hat{\sigma}_t$, N, M and Q are respectively the axial force, the bending moment and the shear force of standard beam theory, whereas M_ϕ and Q_ϕ are an additional bending moment and an additional shear force which are conjugate to the new generalized strains $\frac{\partial \Psi}{\partial x}$ and Ψ , respectively.

The generalized constitutive matrices $\hat{\mathbf{D}}_b$ and $\hat{\mathbf{D}}_t$ are

$$\hat{\mathbf{D}}_p = \iint_A E^k \begin{bmatrix} 1 & -z & \phi^k \\ -z & z^2 & -z\phi^k \\ \phi^k & -z\phi^k & (\phi^k)^2 \end{bmatrix} dA \quad , \quad \hat{\mathbf{D}}_t = \begin{bmatrix} D_s & -\delta \\ -\delta & \delta \end{bmatrix} \quad (2.50a)$$

Miguel Masó Sotomayor

with

$$D_s = \iint_A G^k dA \quad , \quad \delta = D_s - GA \quad (2.50b)$$

In the derivation of the expression for $\hat{\mathbf{D}}_t$ we have used the definition of β^k of equation (2.38).

The generalized constitutive equation can be written as

$$\hat{\boldsymbol{\sigma}} = \begin{Bmatrix} \hat{\boldsymbol{\sigma}}_p \\ \hat{\boldsymbol{\sigma}}_t \end{Bmatrix} = \hat{\mathbf{D}} \hat{\boldsymbol{\varepsilon}} = \hat{\mathbf{D}} \begin{Bmatrix} \hat{\boldsymbol{\varepsilon}}_p \\ \hat{\boldsymbol{\varepsilon}}_t \end{Bmatrix} \quad \text{with} \quad \hat{\mathbf{D}} = \begin{bmatrix} \hat{\mathbf{D}}_p & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{D}}_t \end{bmatrix} \quad (2.51)$$

This formulation does not require a shear correction parameter k_z .

Layer defined generalized constitutive matrix

In the same way as explained in section 2.1.4, defining the generalized constitutive matrix by layers has great advantages when computing resultant stresses. Expressions are modified and had a cheap additional computational cost

$$\hat{\mathbf{D}}_p = \sum_{k=1}^{n_l} [\hat{\mathbf{D}}_p]^k \quad \hat{\mathbf{D}}_t = \sum_{k=1}^{n_l} [\hat{\mathbf{D}}_t]^k \quad (2.52)$$

and each component is detailed

$$\begin{aligned} [\hat{D}_{p11}]^k &= (z_{k+1} - z_k) b_k E^k = b h_k E^k \\ [\hat{D}_{p22}]^k &= \frac{b}{3} (z_{k+1}^3 - z_k^3) E^k \\ [\hat{D}_{p33}]^k &= \frac{b}{3} \frac{(\bar{\phi}^{k+1})^3 - (\bar{\phi}^k)^3}{\bar{\phi}^{k+1} - \bar{\phi}^k} h^k E^k \\ [\hat{D}_{p12}]^k &= -\frac{b}{2} (z_{k+1} - z_k) E^k = -b h_k \bar{z}_k E^k \\ [\hat{D}_{p13}]^k &= \frac{b}{2} (\bar{\phi}^{k+1} + \bar{\phi}^k) h^k E^k \\ [\hat{D}_{p23}]^k &= -b E^k \left((\bar{\phi}^{k+1} + \bar{\phi}^k) \frac{m_2^k}{4} + (\bar{\phi}^{k+1} - \bar{\phi}^k) \left(\frac{m_3^k}{6} - \frac{m_1^k m_2^k}{4} \right) \frac{1}{h^k} \right) \end{aligned} \quad (2.53a)$$

and

$$\begin{aligned}
 [\hat{D}_{t11}]^k &= b(z_{k+1} - z_k)G^k = bh_k G^k \\
 [\hat{D}_{t11}]^k &= bh_k G^k \left(\frac{G}{G^k} - 1 \right)^2 \\
 [\hat{D}_{t12}]^k &= -[\hat{D}_{t11}]^k + GA^k
 \end{aligned} \tag{2.53b}$$

with

$$\begin{aligned}
 m_1^k &= z^{k+1} - z^k \\
 m_2^k &= (z^{k+1})^2 - (z^k)^2 \\
 m_3^k &= (z^{k+1})^3 - (z^k)^3
 \end{aligned} \tag{2.53c}$$

2.2.6 Virtual work expression

The virtual work expression for a distributed load $f_z = q$ is

$$\iiint_V (\delta \varepsilon_x^k \sigma_x^k + \delta \gamma_{xz}^k \tau_{xz}^k) dV - \int_L \delta w q ds = 0 \tag{2.54}$$

The l.h.s. of equation (2.54) contains the internal virtual work performed by the axial and tangential stresses over the beam volume V and the r.h.s. is the external virtual work carried out by the distributed load.

Substituting equations (2.39a,b) into the expression for the virtual internal work and using equations (2.48) and (2.50a)

$$\begin{aligned}
 \iiint_V (\delta \varepsilon_x^k \sigma_x^k + \delta \gamma_{xz}^k \tau_{xz}^k) dV &= \iiint_V (\delta \hat{\varepsilon}_p^T [\mathbf{S}_p^k]^T \sigma_x^k + \delta \hat{\varepsilon}_t^T [\mathbf{S}_t^k]^T \tau_{xz}^k) dV = \\
 &= \int_L (\delta \hat{\varepsilon}_p^T \hat{\sigma}_p + \delta \hat{\varepsilon}_t^T \hat{\sigma}_t) dx
 \end{aligned} \tag{2.55}$$

The virtual work is therefore written as

$$\int_L (\delta \hat{\varepsilon}_p^T \hat{\sigma}_p + \delta \hat{\varepsilon}_t^T \hat{\sigma}_t) dx - \int_L \delta w q dx = 0 \tag{2.56}$$

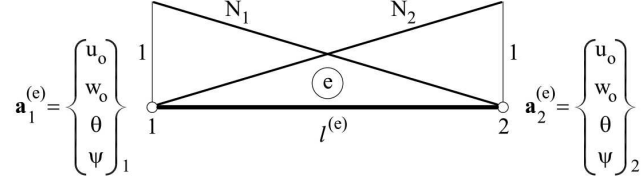


Figure 2.5: Two-noded LRZ composite laminated beam element

2.2.7 Two-noded LRZ beam element

The kinematic variables are u_0, w_0, θ and Ψ . They are discretized using 2-noded linear C^0 beam elements of length $l^{(e)}$ (figure 2.5) as

$$\mathbf{u} = \begin{Bmatrix} u_0 \\ w_0 \\ \theta \\ \Psi \end{Bmatrix} = \sum_{i=1}^2 N_i \mathbf{a}_i^{(e)} = \mathbf{N} \mathbf{a}^{(e)} \quad (2.57)$$

with

$$\mathbf{N} = [N_1 \mathbf{I}_4, N_2 \mathbf{I}_4] \quad , \quad \mathbf{a}^{(e)} = \begin{Bmatrix} \mathbf{a}_1^{(e)} \\ \mathbf{a}_2^{(e)} \end{Bmatrix} \quad , \quad \mathbf{a}_i^{(e)} = \begin{Bmatrix} u_{0_i} \\ w_{0_i} \\ \theta_i \\ \Psi_i \end{Bmatrix} \quad (2.58)$$

where N_i are the standard 1D linear shape functions, $\mathbf{a}_i^{(e)}$ is the vector of nodal DOFs and \mathbf{I}_4 is the 4×4 unit matrix.

Substituting equation (2.57) into the generalized strain vectors of equation (2.39a) gives

$$\hat{\boldsymbol{\varepsilon}}_p = \mathbf{B}_p \mathbf{a}^{(e)} \quad , \quad \hat{\boldsymbol{\varepsilon}}_t = \mathbf{B}_t \mathbf{a}^{(e)} \quad (2.59)$$

The generalized strain matrices \mathbf{B}_p and \mathbf{B}_t are

$$\mathbf{B}_p = [\mathbf{B}_{p1}, \mathbf{B}_{p2}] \quad , \quad \mathbf{B}_t = [\mathbf{B}_{t1}, \mathbf{B}_{t2}] \quad (2.60a)$$

with

$$\mathbf{B}_{p_i} = \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 & 0 & 0 \\ 0 & 0 & \frac{\partial N_i}{\partial x} & 0 \\ 0 & 0 & 0 & \frac{\partial N_i}{\partial x} \end{bmatrix} \quad , \quad \mathbf{B}_{t_i} = \begin{bmatrix} 0 & \frac{\partial N_i}{\partial x} & -N_i & 0 \\ 0 & 0 & 0 & N_i \end{bmatrix} = \begin{bmatrix} \mathbf{B}_{s_i} \\ \mathbf{B}_{\psi_i} \end{bmatrix} \quad (2.60b)$$

where \mathbf{B}_{p_i} and \mathbf{B}_{t_i} are the in-plane and transverse shear strain matrices for node i .

The virtual displacement and the generalized strain fields are expressed in terms of the virtual nodal DOFs as

$$\delta \mathbf{u} = \mathbf{N} \delta \mathbf{a}^{(e)} \quad , \quad \delta \hat{\boldsymbol{\varepsilon}}_p = \mathbf{B}_p \delta \mathbf{a}^{(e)} \quad , \quad \delta \hat{\boldsymbol{\varepsilon}}_t = \mathbf{B}_t \delta \mathbf{a}^{(e)} \quad (2.61)$$

The discretized equilibrium equations are obtained by substituting equations (2.48), (2.57), (2.59) and (2.61) into the virtual work expression (2.56). After simplification of the virtual nodal DOFs, the following standard matrix equation is obtained

$$\mathbf{K} \mathbf{a} = \mathbf{f} \quad (2.62)$$

where \mathbf{a} is the vector of nodal DOFs for the whole mesh.

The stiffness matrix \mathbf{K} and the equivalent nodal force vector \mathbf{f} are obtained by assembling the element contributions $\mathbf{K}^{(e)}$ and $\mathbf{f}^{(e)}$ given by

$$\mathbf{K}^{(e)} = \mathbf{K}_p^{(e)} + \mathbf{K}_t^{(e)} \quad (2.63)$$

with

$$\mathbf{K}_{p_{ij}}^{(e)} = \int_{l^{(e)}} \mathbf{B}_{p_i}^T \hat{\mathbf{D}}_p \mathbf{B}_{p_j} dx \quad , \quad \mathbf{K}_{t_{ij}}^{(e)} = \int_{l^{(e)}} \mathbf{B}_{t_i}^T \hat{\mathbf{D}}_t \mathbf{B}_{t_j} dx \quad (2.64)$$

and

$$\mathbf{f}^{(e)} = \int_{l^{(e)}} N_i q [1, 0, 0, 0]^T dx \quad (2.65)$$

Matrix $\mathbf{K}_p^{(e)}$ is integrated with a one-point numerical quadrature which is exact in this case. In a more compact way, it can be expressed in terms of the generalized strain matrix \mathbf{B}_p and the generalized constitutive matrix $\hat{\mathbf{D}}_p$. Generalized strain matrix \mathbf{B}_p integration requires a one-point quadrature

$$(\mathbf{B}_p^{(e)})_c = \begin{bmatrix} -\frac{1}{l^{(e)}} & 0 & 0 & 0 & \frac{1}{l^{(e)}} & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{l^{(e)}} & 0 & 0 & 0 & \frac{1}{l^{(e)}} & 0 \\ 0 & 0 & 0 & -\frac{1}{l^{(e)}} & 0 & 0 & 0 & \frac{1}{l^{(e)}} \end{bmatrix} \quad (2.66)$$

Full integration of matrix $\mathbf{K}_t^{(e)}$ requires a two-point Gauss quadrature. This however leads to shear locking for slender composite laminated beams. $\mathbf{K}_t^{(e)}$ matrix can be split as Shear locking can be eliminated by reduced integration of all (or some) of the terms of $\mathbf{K}_t^{(e)}$. For this purpose this matrix is split as

$$\mathbf{K}_t^{(e)} = \mathbf{K}_s^{(e)} + \mathbf{K}_\psi^{(e)} + \mathbf{K}_{s\psi}^{(e)} + [\mathbf{K}_{s\psi}^{(e)}]^T \quad (2.67a)$$

with

$$\mathbf{K}_{s_{ij}}^{(e)} = \int_{l^{(e)}} D_s \mathbf{B}_{s_i}^T \mathbf{B}_{s_j} dx \quad , \quad \mathbf{K}_{\psi_{ij}}^{(e)} = \int_{l^{(e)}} \delta \mathbf{B}_{\psi_i}^T \mathbf{B}_{\psi_j} dx \quad (2.67b)$$

$$\mathbf{K}_{s\psi_{ij}}^{(e)} = \int_{l^{(e)}} (-\delta) \mathbf{B}_{s_i}^T \mathbf{B}_{\psi_j} dx$$

and the $\mathbf{K}_t^{(e)}$ matrix exact integration is

$$\mathbf{K}_s^{(e)} = \frac{\hat{D}_s}{l^{(e)}} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \frac{l^{(e)}}{2} & 0 & 0 & -1 & \frac{l^{(e)}}{2} & 0 \\ 0 & \frac{l^{(e)}}{2} & \frac{l^{(e)^2}}{3} & 0 & 0 & -\frac{l^{(e)}}{2} & \frac{l^{(e)^2}}{6} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & -\frac{l^{(e)}}{2} & 0 & 0 & 1 & -\frac{l^{(e)}}{2} & 0 \\ 0 & \frac{l^{(e)}}{2} & \frac{l^{(e)^2}}{6} & 0 & 0 & -\frac{l^{(e)}}{2} & \frac{l^{(e)^2}}{3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.68a)$$

$$\mathbf{K}_\psi^{(e)} = -\frac{\delta}{l^{(e)}} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{l^{(e)^2}}{3} & 0 & 0 & 0 & \frac{l^{(e)^2}}{6} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{l^{(e)^2}}{6} & 0 & 0 & 0 & \frac{l^{(e)^2}}{3} \end{bmatrix} \quad (2.68b)$$

$$\mathbf{K}_b^{(e)} = \frac{\delta}{l^{(e)}} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{l^{(e)}}{2} & 0 & 0 & 0 & -\frac{l^{(e)}}{2} \\ 0 & 0 & 0 & -\frac{l^{(e)^2}}{3} & 0 & 0 & 0 & -\frac{l^{(e)^2}}{6} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{l^{(e)}}{2} & 0 & 0 & 0 & \frac{l^{(e)}}{2} \\ 0 & 0 & 0 & -\frac{l^{(e)^2}}{6} & 0 & 0 & 0 & -\frac{l^{(e)^2}}{3} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.68c)$$

In chapter 4 is made a study of the accuracy of the LRZ beam element for analysis of laminated beams using one and two-point quadratures for integrating $\mathbf{K}_s^{(e)}$, $\mathbf{K}_\psi^{(e)}$

and $\mathbf{K}_{s\psi}^{(e)}$. One-point quadratures integrated matrices are

$$\mathbf{K}_s^{(e)} = \frac{\hat{D}_s}{l^{(e)}} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \frac{l^{(e)}}{2} & 0 & 0 & -1 & \frac{l^{(e)}}{2} & 0 \\ 0 & \frac{l^{(e)}}{2} & \frac{l^{(e)2}}{4} & 0 & 0 & -\frac{l^{(e)}}{2} & \frac{l^{(e)2}}{4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & -\frac{l^{(e)}}{2} & 0 & 0 & 1 & -\frac{l^{(e)}}{2} & 0 \\ 0 & \frac{l^{(e)}}{2} & \frac{l^{(e)2}}{4} & 0 & 0 & -\frac{l^{(e)}}{2} & \frac{l^{(e)2}}{4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.69a)$$

$$\mathbf{K}_\psi^{(e)} = -\frac{\delta}{l^{(e)}} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{l^{(e)2}}{4} & 0 & 0 & 0 & \frac{l^{(e)2}}{4} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{l^{(e)2}}{4} & 0 & 0 & 0 & \frac{l^{(e)2}}{4} \end{bmatrix} \quad (2.69b)$$

$$\mathbf{K}_b^{(e)} = \frac{\delta}{l^{(e)}} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{l^{(e)}}{2} & 0 & 0 & 0 & -\frac{l^{(e)}}{2} \\ 0 & 0 & 0 & -\frac{l^{(e)2}}{4} & 0 & 0 & 0 & -\frac{l^{(e)2}}{4} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{l^{(e)}}{2} & 0 & 0 & 0 & \frac{l^{(e)}}{2} \\ 0 & 0 & 0 & -\frac{l^{(e)2}}{4} & 0 & 0 & 0 & -\frac{l^{(e)2}}{4} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.69c)$$

This beam element is termed LRZ (for **L**inear **T**imoshenko **Z**igzag element).

2.2.8 Shear stresses integration

LRZ results can be much improved by computing τ_{xz} “a posteriori” from the axial stress field using the equilibrium equation ([3], [4])

$$\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xz}}{\partial z} = 0 \quad (2.70)$$

Miguel Masó Sotomayor

The transverse shear stress at a point across the thickness with coordinate z is computed by integrating equation (2.70) as

$$\tau_{xz}(z) = - \int_{h^-}^z \frac{\partial \sigma_x}{\partial x} dz = - \frac{\partial N_z}{\partial x} \quad \text{where} \quad N_z = \int_{h^-}^z \sigma_x dz \quad (2.71)$$

In equation (2.70) N_z is the axial force (per unit width) resulting from the thickness integration of σ_x between the coordinates h^- and z .

Chapter 3

Numerical implementation

3.1 *MAT-fem*. The process

MAT-fem is a *top-down* program and its flow chart is shown in figure 3.1. This simple scheme allows understanding the execution of a Finite Element program.

3.1.1 Start and read input file

The program starts cleaning variables and next ask to the user the name of the input data file (the *m.* extension is not included in the file name). Listing 3.1 shows the first lines of the code corresponding to the variables and the clock set up. Data data is read from a data file as a subroutine.

Input data file

Data file has three groups of variables: the associates wit the section, geometry definition and boundary conditions. In order to simplify the code, the program is free of data validation mechanisms. Even though *MAT-fem* only allows one material, composite sections are allowed, they are treated by vector notation. Listing 3.2 shows the variables associated with the material. Variable `matyp` if the beam has an homogeneous section (`matyp = 1`) or a composite section (`matyp = 2`).

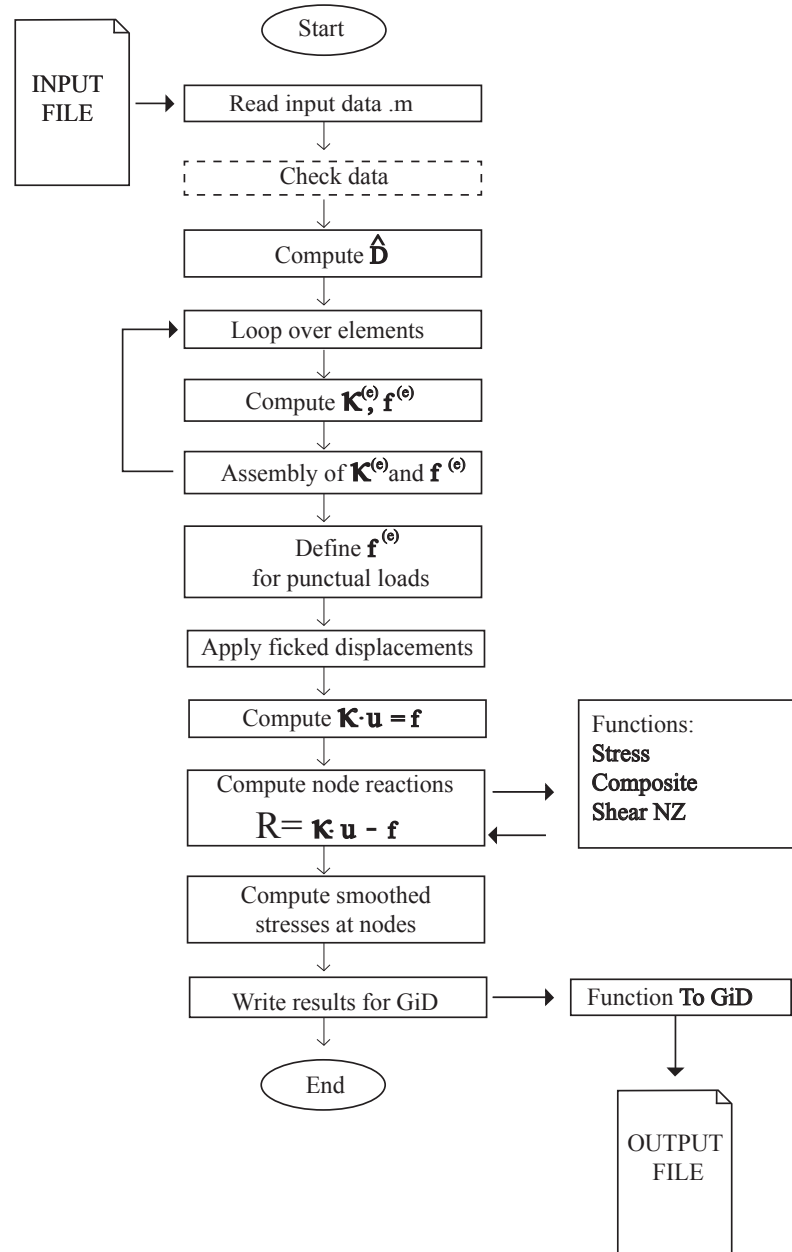


Figure 3.1: *MAT-fem* process flow chart


```

%% 2 Nodes Beam using Timoshenko Theory.
% Composite laminated plane beams
% Reduced integration of K_shear
%
% Clear memory and variables.
clear

file_name = input('Enter the file name : ','s');

tic;                                % Start clock
ttim = 0;                           % Initialize time counter
eval (file_name);                   % Read input file

% Finds basics dimensions
nlayr  = size(young,1);              % Number of layers
npnod  = size(coordinates,1);        % Number of nodes
nelem  = size(elements,1);          % Number of elements
nnode  = size(elements,2);          % Number of nodes por elem
dofpn  = 3;                          % Number of DOF per node
dofpe  = nnode*dofpn;               % Number of DOF per element
nndof  = npnod*dofpn;               % Number of total DOF

ttim = timing('Time needed to read the input file',ttim);

```

Listing 3.1: Program initialization and data reading

```
%  
% Material Properties  
%  
matyp = 2 ;  
young = [  
    1.00e+00    2.190000000e+05 ;  
    2.00e+00    2.190000000e+03 ;  
    3.00e+00    2.190000000e+05 ] ;  
  
poiss = [  
    1.00e+00    2.500000000e-01 ;  
    2.00e+00    2.500000000e-01 ;  
    3.00e+00    2.500000000e-01 ] ;  
  
denss = zeros( 3,2);  
  
hlayr = [  
    1.00e+00    6.667000000e-02 ;  
    2.00e+00    6.666000000e-02 ;  
    3.00e+00    6.667000000e-02 ] ;  
  
blayr = [  
    1.000000000e-01 ;  
    1.000000000e-01 ;  
    1.000000000e-01 ] ;
```

Listing 3.2: Input data file: Material definition

Basic material properties are stored as *n-by-two* array. First column indicates in which layer is set the value specified in the second column. Width is constant over the domain so, it doesn't need to specify the layer. In next section data will be reordered to simplify notation.

Listing 3.3 shows the definition of the coordinates and connectivities by the variables `coordinates` and `elements`.

```
%
% Coordinates
%
global coordinates
coordinates = [
    1.000000000e+00 ;
    5.000000000e-01 ;
    0.000000000e+00 ] ;
%
% Elements
%
global elements
elements = [
    3    ,    2    ;
    2    ,    1    ] ;
```

Listing 3.3: Input data file: Geometry definition

`coordinates` is an array containing nodal x -coordinate. The number of any node corresponds to the position that keeps their coordinates in the array. `elements` defines the number of elements and its connectivities, i -th row corresponds to i -th element.

The last group of variables define the boundary conditions, as it is shown in listing 3.4

`fixnodes` is a matrix where the number of rows corresponds to the number of prescribed DOF and the number of columns describes in the following order the restricted node, the fixed DOF code and the value for this DOF. The `pointload` variable is used to define the punctual loads. Like the previous variables, this is a matrix where the number of rows match the number of loads defined in the problem, and the number of columns corresponds to the number of the loaded node, the direction in which acts and the value of the load. `sideload` is a matrix where the number of rows match the total element number, and the number of columns is the number of DOF's.

```
%  
% Fixed Nodes  
%  
fixnodes = [  
    3 , 1 , 0.000000000e+00 ;  
    3 , 2 , 0.000000000e+00 ;  
    3 , 3 , 0.000000000e+00 ;  
    3 , 4 , 0.000000000e+00 ] ;  
%  
% Point loads  
%  
pointload = [  
    1 , 1 , 0.000000000e+00 ;  
    1 , 2 , -1.000000000e+00 ;  
    1 , 3 , 0.000000000e+00 ] ;  
%  
% Side loads  
%  
uniload = sparse (2 , 2);  
uniload ( 1 , 1) = 0.000000000e+00 ;  
uniload ( 2 , 1) = 0.000000000e+00 ;  
uniload ( 1 , 2) = -1.000000000e+00 ;  
uniload ( 2 , 2) = -1.000000000e+00 ;
```

Listing 3.4: Input data file: boundary conditions definition

```

young = sortrows(young,1);
young = young(:,2);
poiss = sortrows(poiss,1);
poiss = poiss(:,2);
denss = sortrows(denss,1);
denss = denss(:,2);
hlayr = sortrows(hlayr,1);
hlayr = hlayr(:,2);

```

Listing 3.6: Data checking

Zigzag-Timoshenko compatibility

Sometimes can be interesting to analyze the same problem under two beam theories in order to make a comparison. Listing 3.5 shows some code lines that eliminate the ψ condition, corresponding to the 4th DOF.

```

[condnum c] = find(fixnodes(:,2)==4);
condnum = setdiff(1:size(fixnodes,1),condnum);
fixnodes = fixnodes(condnum,:);

```

Listing 3.5: Timoshenko

3.1.2 Generalized constitutive matrix

The program main's purpose is to demonstrate the implementation of the FEM, so some simplifications are made, like use a unique section along the whole beam. Constitutive matrix will not vary element to element and it is evaluated before initializing the stiffness matrix.

The first step is order data input from GiD. Algorithm is shown in listing 3.6. Once values are sorted, each row of the variable stores the corresponding layer value.

Before computing the generalized constitutive matrix, some values are set. Listing 3.7 gives an example of basic data setting of the laminated section. Variable **hsect** stores the beam depth and variable **shear** is a n_{layer} -rows vector containing the shear moduli.

Auxiliary variables as vertical coordinate (**zlayr**), middle point coordinate (**mlayr**) and self-weight (**weight**) are computed in this section.

```
hsect = sum(hlayr);
shear = young./(2*(1+poiss));

zlayr = zeros (nlayr + 1,1);
mlayr = zeros (nlayr,1);
weight= 0;
for k = 1 : nlayr
    zlayr(k + 1) = zlayr(k) + hlayr(k);
    mlayr(k) = zlayr(k) + hlayr(k)/2;
    weight    = weight + hlayr(k)*blayr(k)*denss(k);
end
zlayr = zlayr - hsect/2;
mlayr = mlayr - hsect/2;
```

Listing 3.7: Basic section values definition

```
for ilayr = 1 : nlayr
    weight    = weight + hlayr(ilayr)*blayr(ilayr)*denss(ilayr);
    D_la(ilayr) = hlayr(ilayr)*blayr(ilayr)*young(ilayr);
    D_lb(ilayr) = 1/3*(zlayr(ilayr+1)^3 - zlayr(ilayr)^3)* ...
                    blayr(ilayr)*young(ilayr);
    D_lab(ilayr) = -hlayr(ilayr)*blayr(ilayr)*mlayr(ilayr)* ...
                    young(ilayr);
    G_l(ilayr)   = hlayr(ilayr)*blayr(ilayr)*shear(ilayr);
end
```

Listing 3.8: Timoshenko layer defined constitutive matrix

Timoshenko constitutive matrix

To compute the generalized constitutive matrix, first is calculated by layers (listing 3.8) and then, generalized constitutive matrix is obtained by summing terms (listing 3.9)

This variables had great advantages when computing resultant stresses.

Shear correction parameter Since composite laminated Timoshenko theory assumes a uniform distribution of shear stresses, it needs a correcting shear parameter depending on the section properties. Shear correction parameter increases deflection in very heterogeneous sections.

To use an easier formulation (equation 2.11) listing 3.10 implements equation 2.12.

```

D_mata = sum(D_la);           % Section stiffness
D_matb = sum(D_lb);
D_matab = sum(D_lab);
G_mat = sum(G_l);
D_mat = D_mata * D_matb - D_matab^2;

```

Listing 3.9: Timoshenko generalized constitutive matrix

```

dnaxs = -D_matab/D_mata;      % Axis translation
znaxs = zlayr - dnaxs;
mnaxs = mlayr - dnaxs;

```

Listing 3.10: Axis translation to neutral axis

Listing 3.11 shows the implementation of shear correction parameter (equation 2.21)

Variables **S1**, **S2** and **S3** are auxiliary variables used to compute the integration of the static moment of the Young modulus with respect to the coordinate z (equation 2.17)

Zigzag constitutive matrix

In this section generalized constitutive matrix is also computed by layers, but before compute the stiffness, zigzag function must be set (listing 3.12). The zigzag function adds an additional deformation pattern, that is coupled with axial, bending and shear patterns of deformation. Scalar variable **G** is the averaged shear stiffness. Vector variables **beta** and **phi** are the derivative of the zigzag function and the zigzag function respectively.

D_lp and **D_lt** variables store the in-plane and transverse stiffness respectively for each layer. Both are three dimensional arrays. **D_matp** and **D_matt** variables store the generalized constitutive matrix, they are computed as the sum of **D_lp** and **D_lt** variables along the layers.

D_matp and **D_matt** variables (listing 3.14) store the generalized constitutive matrix, they are computed as the sum of **D_lp** and **D_lt** variables along the layers.

```

S1 = 0;                                % Compute the shear correction parameter
S3 = 0;
for jlayr = 1 : nlayr
    if jlayr > 1
        S1 = S1 + young(jlayr-1)*(zlayr(jlayr)^2 -zlayr(jlayr-1)^2);
    end
    S2 = 1/4*(S1-young(jlayr)*zlayr(jlayr)^2)^2 ...
        *(zlayr(jlayr+1) -zlayr(jlayr) )+ ...
        1/6*(S1-young(jlayr)*zlayr(jlayr)^2) *young(jlayr) ...
        *(zlayr(jlayr+1)^3-zlayr(jlayr)^3)+ ...
        1/20*young(jlayr)^2*(zlayr(jlayr+1)^5-zlayr(jlayr)^5);
    S3 = S3 + blayr(jlayr)/shear(jlayr)*S2 ;
end
kz = D_naxb^2/G_mat/S3;                % Shear correction parameter
D_ls = kz*G_l;                          % Layer shear stiffness
D_mats = kz*G_mat;                      % Section shear stiffness

```

Listing 3.11: Shear correction parameter computing

```

G = 0;
for k = 1 : nlayr
    G = G + hlayr(k)/shear(k);
end
G = hsect/G;

beta = zeros(nlayr,1);
phi = zeros(nlayr+1,1);
for k = 1 : nlayr
    beta(k) = G/shear(k) - 1;
    phi(k+1)= phi(k) + beta(k)*hlayr(k);
end

```

Listing 3.12: Zigzag function


```

for k = 1 : nlayr
    D_lp(1,1,k) = young(k)*blayr(k)*hlayr(k);
    D_lp(2,2,k) = young(k)*blayr(k)*dz3(k)/3;
    D_lp(3,3,k) = young(k)*blayr(k)*hlayr(k)*...
        (phi(k+1)^3-phi(k)^3)/(phi(k+1)-phi(k))/3;
    D_lp(1,2,k) = -young(k)*blayr(k)*hlayr(k)*mlayr(k);
    D_lp(1,3,k) = young(k)*blayr(k)*hlayr(k)*(phi(k+1)+phi(k))/2;
    D_lp(2,3,k) = -young(k)*blayr(k)*((phi(k+1)+phi(k))*dz2(k)/4..
        + (phi(k+1)-phi(k))*(dz3(k)/6-mlayr(k)*dz2(k)/4)/hlayr(k));
    D_lp(2,1,k) = D_lp(1,2,k);
    D_lp(3,1,k) = D_lp(1,3,k);
    D_lp(3,2,k) = D_lp(2,3,k);

    D_s = hlayr(k)*blayr(k)*shear(k);
    D_lt(1,1,k) = D_s;
    D_lt(2,2,k) = D_s * beta(k)^2;
    D_lt(1,2,k) = G*blayr(k)*hlayr(k) - D_s;
    D_lt(2,1,k) = D_lt(1,2,k);
end

```

Listing 3.13: Generalized constitutive matrix by layers

```

D_matp = sum(D_lp,3);           % In-plane section stiffness
D_matt = sum(D_lt,3);           % Transverse section stiffness

```

Listing 3.14: Generalized constitutive matrix

```
% Dimension the global matrices.
StifMat = sparse ( nn dof , nn dof ); % Create the stiffness matr
force   = sparse ( nn dof , 1 );      % Create the force vector
```

Listing 3.15: Global matrix initialization

```
% Element cycle.
for ielem = 1 : nelem

    lnods(1:nnode) = elements(ielem,1:nnode);

    coor_x(1:nnode) = coordinates(lnods(1:nnode),1); % Elem X coor

    len = coor_x(2) - coor_x(1); % x_j > x_i

    const = D_mata/len;

    K_axial = [ 1 , 0 , 0 , -1 , 0 , 0 ;
                0 , 0 , 0 , 0 , 0 , 0 ;
                0 , 0 , 0 , 0 , 0 , 0 ;
                -1 , 0 , 0 , 1 , 0 , 0 ;
                0 , 0 , 0 , 0 , 0 , 0 ;
                0 , 0 , 0 , 0 , 0 , 0 ];
```

Listing 3.16: Element cycle beginning to evaluate and assemble the elemental stiffness matrix

3.1.3 Elemental stiffness matrix and its assemble

The code shown in listing 3.15 defines the global stiffness matrix and the equivalent nodal forces vector a respectively as a **sparse** matrix and vector. This sparse indexing optimizes the memory using the MATLAB's tools.

Another simplification is made in this section to allow *MAT-fem* to demonstrate the FEM. This routine is free of local axes, so global matrices are directly integrated. Listing 3.16 shoes the elemental loop in which the program calculates and assembles the stiffness matrix and the equivalent nodal load vector for each element. The cycle begins recovering the geometric properties of each element. In the vector **lnods** the element's nodal connectivity are stored and in the **coord** matrix the coordinates from these nodes are keep.

Next step is to calculate the elemental stiffness matrix. The simple beam element allows defining directly the stiffness matrices, listing 3.17 shows the code of reduced

```

const      =  D_mats/len;

K_shear = [  0 ,    0 ,    0 ,    0 ,    0 ,    0 ;
            0 ,    1 ,   len/2 ,    0 ,   -1 ,   len/2 ;
            0 ,  len/2 , len^2/4 ,    0 , -len/2 , len^2/4 ;
            0 ,    0 ,    0 ,    0 ,    0 ,    0 ;
            0 ,   -1 ,  -len/2 ,    0 ,    1 ,  -len/2 ;
            0 ,  len/2 , len^2/4 ,    0 , -len/2 , len^2/4];

K_shear = K_shear * const;

K_elem = K_axial + K_bend + K_axbn + K_axbn' + K_shear;

```

Listing 3.17: Shear stiffness matrix integration and elemental stiffness matrix

```

B_matp = [-1/len,  0 ,    0 ,    0 , 1/len,  0 ,    0 ,    0 ;
          0 ,    0 ,-1/len,    0 ,    0 ,    0 , 1/len,    0 ;
          0 ,    0 ,    0 ,-1/len,    0 ,    0 ,    0 , 1/len];

K_p = len * B_matp' * D_matp * B_matp;

```

Listing 3.18: In-plane stiffness integration

integration of the shear matrix from equation 2.34 in Timoshenko theory. Listing 3.18 shows the code of the in-plane stiffness matrix integration from zigzag theory, deduced from equation 2.66

Since `uniload` matrix is defined `nelem×2` (listing 3.4), while the elemental loop is running, the uniform distributed loads are extrapolated to the equivalent nodal forces and stored in `ElemFor` vector, this code is shown in listing 3.19. The nodal contribution is evaluated equality on each node due to linear function form.

```

f1      = (      uniload(ielem,1))*len/2;
f2      = (-weight + uniload(ielem,2))*len/2;
ElemFor = [ f1, f2, 0 , 0 , f1, f2, 0 , 0 ];

```

Listing 3.19: Mass and uniform load vector

Finally, before stiffness matrix and force vector assembly the `eqnum` is defined (listing 3.20). This variable stores the global equation number for all the DOF's involved in the element. The force vector needs one cycle from 1 to `dofpe` (number of equations per element), while the stiffness matrix requires two cycles from 1 to `dofpe`. With this scheme the elemental matrices and vectors are stored tempo-

rally.

```
% Finds the equation number list for the i-th element
for i=1:nnode
    ii = (i-1)*dofpn;
    for j =1:dofpn
        eqnum(ii+j) = (lnods(i)-1)*dofpn+j; % Build equation numbe
    end
end

% Assemble the force vector and the stiffness matrix
for i = 1 : dofpe
    ipos = eqnum(i);
    force (ipos) = force(ipos) + ElemFor(i);
    for j = 1 : dofpe
        jpos = eqnum(j);
        StifMat (ipos,jpos) = StifMat (ipos,jpos) + K_elem(i,j);
    end
end

end % End element cycle
```

Listing 3.20: Elemental stiffness matrix assembling

When the elemental cycle stops the global stiffness matrix is assembled and in next steps, punctual loads and Dirichlet conditions are added before solving the global system.

3.1.4 External loads

Adding uniform loads is explained in previous section. Punctual loads calculation (listing 3.21) is simpler than uniform loads calculation. It's calculation is made by adding the value of the load to the equivalent nodal force vector. This calculation needs a loop over the number of point loads and a local to global equation number conversion.

```
%% Add point loads conditions to the force vector
for i = 1 : size(pointload,1)
    ieqn = (pointload(i,1)-1)*dofpn+pointload(i,2); % Finds eq num
    force(ieqn) = force(ieqn) + pointload(i,3); % add the forc
end
```

Listing 3.21: Equivalent nodal force vector for a punctual load

3.1.5 Fixed displacements

Listing 3.22 shows the code which defines the loop over the prescribed DOF to assign to the displacements vector **u** the known values defined by **fixnodes**. Also the fix vector is defined to keep the equation numbers of the restricted DOF.

Finally the force vector is modified into **forceDC** vector with the product of the **StifMat** matrix and the **u** vector, which at this moment contains only the values of those DOF which have been restricted. Conserving **force** vector without being affected by Dirichlet conditions is needed to compute the reactions.

```
for i = 1 : size(fixnodes,1)
    ieqn = (fixnodes(i,1)-1)*dofpn+fixnodes(i,2); % Finds eq num
    u (ieqn) = fixnodes(i,3); % and store the solution in u
    fix(i) = ieqn; % and mark the eq as a fix value
end

forceDC = force - StifMat * u; % adjust the rhs
```

Listing 3.22: Adding prescribed values

3.1.6 Solution of the equation system

The strategy used in MAT-fem consists of solving the global equation system (listing 3.23) without considering those DOF whose values are known. The **FreeNodes** vector contains the list of the equations to solve.

The **FreeNodes** vector is used as a DOF index and allows us to write in a simple way the solution to the equations system. MATLAB takes care to choose the most suitable algorithm to solve the problem, being totally transparent for the user the solution of the system.

```
% Compute the solution by solving StifMat * u = force for the
% remaining unknown values of u.
FreeNodes = setdiff ( 1:nndof, fix ); % Finds the free node list

u(FreeNodes) = StifMat(FreeNodes,FreeNodes) \ forceDC(FreeNodes);
```

Listing 3.23: Solution of the global equation system

3.1.7 Reactions

The solution to the equations system (listing 3.24) is in the **u** vector, therefore the reaction calculus is made by means of the expression $R = \text{StifMat} * u - \text{force}$. Note that in this calculus, is used the **force** vector without the Dirichlet conditions contribution. It is obvious that the reactions on the unprescribed nodes is zero.

```
%% Compute the reactions on the fixed nodes as R = StifMat * u - F
reaction = sparse(nndof,1);
reaction(fix) = StifMat(fix,1:nndof) * u(1:nndof) - force(fix);
```

Listing 3.24: Computing reactions

3.1.8 Strains, stresses and resultant stresses

Once the nodal displacements have been found it is possible to evaluate the resultant stresses in the elements by the **DBu** expression. The deformation matrix **B** is calculated at the integration points, so the stresses are referred to these points. In order to transfer the values of the stresses at the integration points towards the element nodes it is necessary to review in detail and in a later section the calculation of these values. Listing 3.25 presents the subroutine call for the nodal stresses evaluation which are store in the **StrNod** matrix.

```
%% Compute the stresses
StrNod = Stress_Beam_LRZ_v1_1(D_matp,D_matt,u);
```

Listing 3.25: Calling stresses evaluation

A lot of information remains stored in a composite laminated beam element, so *MAT-fem* calls an specific subroutine to calculate the strains, stresses and resultant stresses along the beam thickness. **Composite_Beam** subroutine (listing 3.26) computes this values. The subroutine structure is like **Stress_Beam** subroutine (listing 3.25), so it is also discussed in next section.

```
[lDspNod lStrNod lResStrGP] = Composite_Beam_LRZ_v1_1 ...
                             (young,shear,zlayr,phi,beta,D_lp,D_lt,u);

[shrStr coor_z] = ShearNz_Beam_LRZ_v1_1 (young,zlayr,lStrNod);
```

Listing 3.26: Strains, stresses and resultant stresses evaluation along the beam thickness

Shear stresses are computed from the kinematic variables interpolated by the FEM even though, as explained in section 2.2.8 zigzag theory can integrate shear stresses from the constitutive equation (2.70). Listing 3.26 shows the code calling `ShearNz_Beam_LRZ` subroutine to integrate shear stresses.

Label LRZ refers to zigzag theory (from LRZ element) and subroutines related to Timoshenko theory are labeled with `Timoshenko`.

3.1.9 More about stresses evaluation

`Stress_Beam` subroutine computes stresses on Gauss points and project the stresses toward the nodes. In C^0 beam shape functions the stresses are constant and the nodal extrapolation is trivial.

Listing 3.27 shows the subroutine initialization and the shown code sets the basic variables. The subroutine needs the generalized constitutive matrix and the displacements. Additionally the nodal coordinates and elements connectivities will be used, as they are defined as global variables.

The `StrNod` matrix is initialized to zeros to store the value of the nodal stresses, in the last column the number of elements that concur in the node are counted. This is necessary to make a nodal stress mean.

```
function StrNod = Stress_Beam_LRZ_v1_1 (D_matp,D_matt,u)
%% Evaluates the resultant stresses at the gauss point and smooth
%    the values to the nodes.

    global coordinates;
    global elements;

    nelem = size(elements,1);           % Number of elements
    nnode = size(elements,2);          % Number of nodes por elem
    npnod = size(coordinates,1);       % Number of nodes
    StrNod = zeros(npnod , 6 );         % Create array for stresses
    dofpn = 4;                         % Number of DOF per node
    dofpe = dofpn*nnode;               % Number of DOF per element
    eqnum = zeros(dofpe);              % Equation number list
```

Listing 3.27: Subroutine initialization

Like in the stiffness matrix, the stress evaluation requires a loop over the elements, recovering the element's connectivities (`lnods`), coordinates for these nodes (`coor_x`) as well as the displacements in `u_elem` as it is shown in listing 3.28

```
% One gauss point for stress evaluation
gaus0 = 0.0; % One Gauss point for stresses evaluation

B_matt = [ 0,-1/len,-(1-gaus0)/2, 0, 0, 1/len,-(1+gaus0)/2, 0;
           0, 0, 0, (1-gaus0)/2, 0, 0, 0, (1+gaus0)/2];
```

Listing 3.29: Strain matrices evaluation on Gauss points

```
% Element cycle.
for ielem = 1 : nelem

    lnods(1:nnode) = elements(ielem,1:nnode);

    coor_x(1:nnode) = coordinates(lnods(1:nnode),1); % Elem X coor

    % Finds the equation number list for the i-th element
    for i=1:nnode
        ii = (i-1)*dofpn;
        for j =1:dofpn
            eqnum(ii+j) = (lnods(i)-1)*dofpn+j; % Build equation numbe
        end
    end

    % Recover the nodal displacements for the i-th element
    u_elem(1:dofpe)=u(eqnum(1:dofpe));
```

Listing 3.28: Recovering the element coordinates and nodal displacement

Inside the elemental loop strain matrix **B** is defined and evaluated on the integration point (listing 3.29). Element stresses are stored in **Str_p_g0** and **Str_t_g0** arrays, this values are accumulated in the **StrNod** where the last column is the number of elements that share the node to evaluate later the nodal average (listing 3.30).

StrNod is a $\text{npnod} \times \text{by} \times 4$ array in Timoshenko theory; corresponding to the number of nodes and the number of stresses (equation (2.6) defines N , M and Q resultant stresses, and the additional index to store the number of entries). In zigzag theory **StrNod** is a $\text{npnod} \times \text{by} \times 6$ array due to the two additional stresses M_ϕ and Q_ϕ defined in equation (2.48).

```
Str_p_g0 = D_matp * B_matp * transpose(u_elem);
Str_t_g0 = D_matt * B_matt * transpose(u_elem);

StrNod(lnods(1),1) = StrNod(lnods(1),1)+Str_p_g0(1);
```



```

StrNod(lnods(2),1) = StrNod(lnods(2),1)+Str_p_g0(1);
.
.
.
StrNod(lnods(1),6) = StrNod(lnods(1),6)+1;
StrNod(lnods(2),6) = StrNod(lnods(2),6)+1;

```

Listing 3.30: Stresses evaluation and projection toward nodes

Finally listing 3.31 code defines a loop over nodes to compute the nodal stresses mean.

```

for i = 1 : npnod
    StrNod(i,1) = StrNod(i,1)/StrNod(i,4);
    StrNod(i,2) = StrNod(i,2)/StrNod(i,4);
    StrNod(i,3) = StrNod(i,3)/StrNod(i,4);
end

```

Listing 3.31: Nodal stresses mean

3.1.10 More about thickness distribution of strains and stresses

In `Composite_Beam` subroutine (listing 3.26) a more detailed calculus of strains stresses and resultant stresses is performed. Since strains doesn't need the evaluation of the **B** matrix it are computed directly on the nodes. Even though stresses and resultant stresses need to be evaluated at the integration points.

```

function [lDspNod lStrNod lResStrGP] = Composite_Beam_LRZ_v1_1 ...
    (young,shear,zlayr,phi,beta,D_lp,D_lt,u)

    global coordinates;
    global elements;

```

Listing 3.32: Subroutine initialization

In listing 3.32 the initialization of the subroutine is shown. It is initialized with the layer defined generalized constitutive matrix rather the simple generalized constitutive matrix, and it needs some extra information as the young and shear moduli, geometrical section parameters and the zigzag function.

```

lDspNod = zeros(npnod,4,nlayr+1); % Create array for displaceme
lStrNod = zeros(npnod,3,2*nlayr); % Create array for stresses

```

```
lResStrGP = zeros(nelem,5,nlayr); % Create array for res stress
```

Listing 3.33: Set variables

Listing 3.33 shows the variables definition. `lDspNod` array stores the displacements field. The nodal reference is indexed in the first dimension, the displacement interpolation (three in Timoshenko theory, equation (2.27), and four in zigzag theory, equation (2.27)) is indexed in the second dimension and the thickness position is indexed in the third dimension.

`lStrNod` array stores the stresses field. The nodal reference is indexed in the first dimension, the stress reference is indexed in the second dimension (two for σ and τ , and the third to store the number of entries) and the thickness position is indexed in the third dimension. The stresses are a discontinuous field, so they need two values on each layer, on the lower and the upper boundary.

`lResStrGP` array stores the layer's contribution to the resultant stresses. Once they are computed, they remain on the Gauss points. `lResStrGP` is analog to `StrNod`, but changing nodes by elements.

```
% Compute the layer displacements
% Finds the equation number list for the i-th node
for nn = 1 : npnod
    ndof1(nn) = (nn-1)*dofpn + 1;
    ndof2(nn) = (nn-1)*dofpn + 2;
    ndof3(nn) = (nn-1)*dofpn + 3;
    ndof4(nn) = (nn-1)*dofpn + 4;
end
```

Listing 3.34: Finding the DOF reference for each node

Listing 3.34 shows the code computing the equation number to evaluate in listing 3.35 the displacements field, from equation (2.1) in Timoshenko theory and equation (2.36a) in zigzag theory.

```
for k = 1 : nlayr + 1
    lDspNod(:,1,k) = u(ndof1) - zlayr(k)*u(ndof3) + phi(k)*u(ndof4);
    lDspNod(:,2,k) = u(ndof2);
    lDspNod(:,3,k) = u(ndof3);
    lDspNod(:,4,k) = u(ndof4)*phi(k);
end
```

Listing 3.35: Computation of the displacement field

```
function [shrStr coor_z] = ShearNz_Beam_LRZ_v1_1 ...
    (young,zlayr,lStrNod)
% This script integrates shear stress from constitutive eq.

global coordinates;
global elements;

npz      = 50;                                % number of points per layer
shrStr = zeros(npnod,npz*nlayr,2); % Create array for shear str
```

Listing 3.38: Subroutine initialization

Stresses are evaluated by the \mathbf{DS} expression from equation (2.5) in Timoshenko theory and from equation (2.41a) in zigzag theory. Code from listing 3.36 gives an example how to evaluate the \mathbf{DS} product at different beam depth positions.

```
for k = 1 : nlayr
    E_Sp(2*(k-1)+1,:) = young(k)*[ 1 , -zlayr( k ), phi( k ) ];
    E_Sp(2*(k-1)+2,:) = young(k)*[ 1 , -zlayr(k+1), phi(k+1) ];
    G_St(2*(k-1)+1,:) = shear(k)*[ 1 , beta(k)];
    G_St(2*(k-1)+2,:) = shear(k)*[ 1 , beta(k)];
end
```

Listing 3.36: Constitutive matrix and strain-displacement transformation matrix

Once \mathbf{DS} product is evaluated, stresses can be computed inside an elemental loop substituting equation (2.28) on equation (2.5), and it are stored on $\mathbf{lStrNod}$, in the same way that \mathbf{StrNod} .

```
Str1_g0 = E_Sp*B_matp*transpose(u_elem);
Str2_g0 = G_St*B_matt*transpose(u_elem);
```

Listing 3.37: Stresses evaluation

Shear stresses integration. Section 2.2.8 explains how to integrate the thickness distribution of shear stresses. Listing 3.38 shows the basic parameters of the subroutine to evaluate shear stresses.

Vertical coordinate is split on several points where shear stresses distribution will be evaluated. `npz` variable defined on listing code 3.38 sets the number of points to evaluate the stresses (listing 3.39 and 3.40).

```
%% Vertical coordinate
coor_z = zeros(npz*nlayr,1);
for k = 1 : nlayr
    coor_z(npz*(k-1)+1:npz*k) = linspace(zlayr(k),zlayr(k+1),npz);
end
```

Listing 3.39: Split vertical coordinate

```
diffNz = (Nz(lnods(2),:) - Nz(lnods(1),:))/len;

shrStr(lnods(1),:,1) = shrStr(lnods(1),:,1) - diffNz;
shrStr(lnods(2),:,1) = shrStr(lnods(2),:,1) - diffNz;
```

Listing 3.41: Stresses derivative

MATLAB commands allow integrating shear stresses in a few lines. Two loops over nodes and layers are request.

```
%% Normal stresses integration
for i = 1 : npnod
    for k = 1 : nlayr
        Nz(i,npz*(k-1)+1:npz*k) = ...
            linspace(lStrNod(i,1,2*(k-1)+1),lStrNod(i,1,2*(k-1)+2),npz);
    end
    Nz(i,:) = cumtrapz(coor_z,Nz(i,:));
end
```

Listing 3.40: Stresses evaluation and integration along the beam depth

Finally, shear stresses are obtained with the x derivative of the Nz variable. Shear stresses derivative is obtained with the same strategy than stresses. Listing 3.41 makes the derivative inside the element and store the values on `shrStr` variable.

3.1.11 Writing for postprocessing

Once calculated the nodales displacements, the reactions and the stresses it is come to overturn these values to the postprocess files from where GiD will be able to present/display them in a graphical way. This is made in the subroutine ToGiD shown in listing 3.42

```
%% Graphic representation.
```

```
ToGiD_Beam_LRZ_v1_1(file_name,zlayr,u,reaction,...
                    StrNod,lDspNod,lStrNod,lResStrGP);
```

Listing 3.42: Postprocess call

3.2 Graphical User Interface

MAT-fem is named by GiD as a *Problem Type* [7]. In this section the Graphical User Interface (GUI) implemented in GiD is reviewed. In order to accede to GUI is necessary to locate `MAT-fem_beams.gid` folder inside the GiD installation directory. Then user is able to select from the Data menu in the option Problemtyp the module corresponding to *MAT-fem beams*. When selected, the image shown in figure 3.2 will appear.



Figure 3.2: MAT-fem beams GUI start up

3.2.1 Preprocess

Solving a problem with *MAT-fem* is a very simply thing once geometry have been defined. Holding an line elements mesh brings preprocess to be early nonexistent. The user just needs to follow the icons of the *MAT-fem*'s graphical menu.



The first button is for identify those geometric elements that present movements restriction. When pressing on, an emergent window (figure 3.3) will appear to select the nodes or lines were the displacements are restricted. The check boxes identify the fixed direction. Also it is possible to give a non-zero value to constrain.

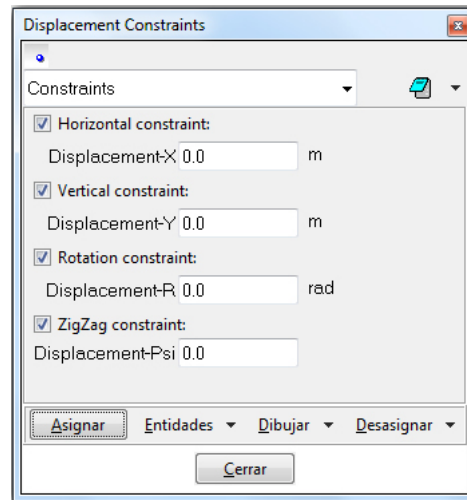


Figure 3.3: Fixed displacement condition



The icon button is used for punctual loads allocation. When selecting, an emergent window (figure 3.4) allows giving the load value in the global coordinate system. Once it is defined is necessary to select the nodes were to applied it.

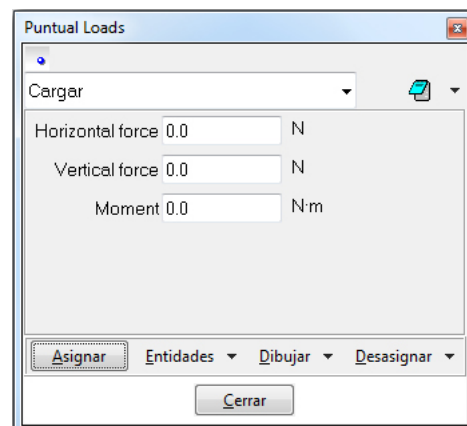


Figure 3.4: Point loads assignation



The icon associated to the uniformly distributed loads permits to assign this condition on the geometry's lines. The emergent window (figure 3.5) allows introducing the value of the load by length unit referred the global axes system.



The material properties definition is made trough the following button which leads to a new emergent window (figure 3.6) to define the material variables. This assignation is used with the classical beam theories: Euler-Bernuolli and

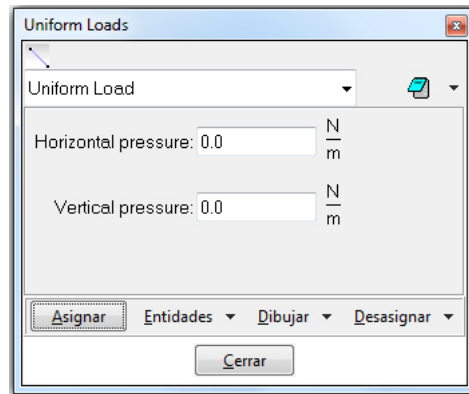


Figure 3.5: Uniform loads assignment

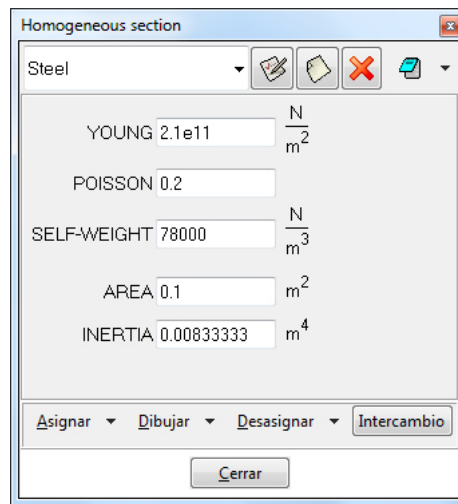


Figure 3.6: Material properties definition

Timoshenko. As was mention before, in MAT-fem, for simplify reasons only one material is allowed.



Composite material definition is made by this alternative icon which open an emergent window (figure 3.7). Each field is defined by the layer assignation and the corresponding property value. Obviously the number of layers must be the same in all the fields. Note that this section properties assignation mode is equivalent to a unique material. A composite section must be processed with the composite laminated plane beams Timoshenko theory or even with the zigzag theory.



The general properties button allows acceding to the window shown in figure 3.8 where the title of the problem is identified and the gravity forces can be

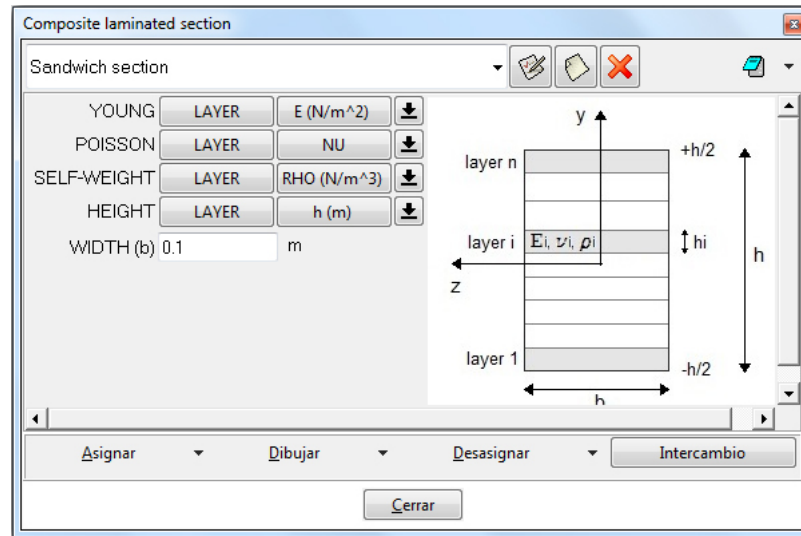


Figure 3.7: Composite material definition

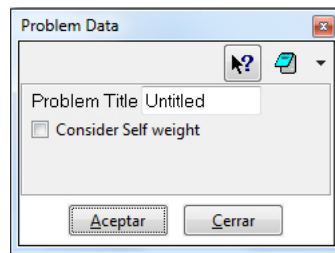


Figure 3.8: General properties of the problem

considered.



Once defined the boundary conditions and the material properties it is necessary to make the domain discretization. The mesh button is used to create the mesh.



The data file writing is made when pressing the last button shown in the menu. All the geometrical properties of the problem, as well as the material are written to the data file in the specific reading format for MAT-fem. MATLAB .m extension must be added to the file name.

Details about the configuration files

GiD allows creating a problem type through several files. All of them are named `problem_type_name.extension`:

- `MAT-fem_beams.cnd` Conditions definitions
- `MAT-fem_beams.mat` Materials properties
- `MAT-fem_beams.prb` Problem and intervals data
- `MAT-fem_beams.uni` Units system
- `MAT-fem_beams.sim` Conditions symbols
- `MAT-fem_beams.tcl` Tcl extension
- `MAT-fem_beams.bas` Information for the input data file

The GUI icons are created by the `.tcl` file [7], [8]. In the present work is added the *composite materials* assignment icon, it calls a materials book with composite laminated sections (listing 3.43).

```
proc MyBitmaps { dir { type "DEFAULT INSIDELEFT"} } {
    global MyBitmapsNames MyBitmapsCommands \
        MyBitmapsHelp MAT-fem_Beams
    set MyBitmapsNames(0) "fix.gif pload.gif unload.gif \
        material_homog.gif material_comp.gif units.gif \
        mesh.gif write.gif"
    set MyBitmapsCommands(0) [list \
        [list -np- GidOpenConditions "Displacement_Constraints"] \
        [list -np- GidOpenConditions "Puntual_Loads"] \
        [list -np- GidOpenConditions "Uniform_Loads"] \
        [list -np- GidOpenMaterials "Homogeneous_section"] \
        [list -np- GidOpenMaterials "Composite_laminated_section"] \
        [list -np- GidOpenProblemData] \
        "Meshing generate" \
        "File WriteCalcFile" ]
}
```

Listing 3.43: Tcl file: Commands definitions

Composite laminated sections It keep the same structure than homogeneous materials, but treating properties as vectors rather than scalars. Additionally, is set a hidden property to identify if the section, 1 means homogeneous and 2 means composite. Listing 3.44 shows the code setting the identifier and defining the Young modulus as a variable field.

```
NUMBER: 5 MATERIAL: Sym_3_layered
QUESTION: Identifier
VALUE: 2
STATE: hidden
QUESTION: YOUNG(LAYER,_E_(N/m^2)__)
VALUE: #N# 6 1 2.19e5 2 2.19e3 3 2.19e5
```

Listing 3.44: Materials file: section identifier and definition of the materials properties

Conditions file This file defines the fixed displacements, the punctual loads and the uniform distributed loads. Composite laminated Timoshenko theory introduces an axial displacement so, an axial constraint and axial loads can be defined. Even though zigzag theory only needs a zigzag constraint.

The template file It is the same as the .bas file, it describes the format and structure of the required input data file for the solver. As mentioned before, the input data file has three sections: section properties, geometrical definition and boundary conditions. Section properties and boundary conditions had different writing format: one for the homogeneous case and another for the composite case. The code shown in listing 3.45 sets the `id` variable and prints the values for the homogeneous section case (`id==1`).

```
%
% Material Properties
%
*loop materials
*set var id = MatProp(Identifier,int)
*if(id==1)
*format "    young    = %17.9e ;"
*MatProp(2)
```

Listing 3.45: Template file: section identification

The composite section follows a different structure (listing 3.46). The `id` variable will be used latter to define the format of the conditions file.

```
*elseif(id==2)
*set var C=2
*set var N=MatProp(2,int)
*if(N==MatProp(3,int)&&N==MatProp(4,int)&&N==MatProp(5,int))
young = [
```

```

*for(i=1;i<=N-C;i=i+C)
*format " %6.2e %17.9e "
*MatProp(2,*i) *MatProp(2,*operation(i+1)) ;
*end for
*format " %6.2e %17.9e "
*MatProp(2,*operation(N-1)) *MatProp(2,*N) ] ;

```

Listing 3.46: Template file: composite section format

3.2.2 Postprocess

Once concluded the problem execution in MATLAB, is necessary to return to GiD and open the file postprocess to analyze the obtained results. It is necessary to open any of the generated files that contain the extension *.post.msh or *.post.res. If they are saved inside the project directory GiD will read them automatically.

The obtained results visualization can be done in a diversity forms, due the GiD's graphical possibilities which permits to show the results like a colors gradient, iso-lines, cuts and graphs; allowing the simple interpretation of the obtained results. In figure (3.9) there is an iso-areas example showing the horizontal displacement using the deformed mesh. The example is a thick cantilever beam under an end point.

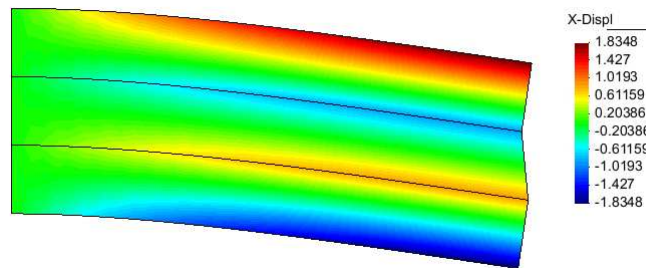


Figure 3.9: Beam postprocess

ToGiD_Beam subroutine writes two different analysis on the postprocess files: the first one prints the classical beam theories results and the second one shows a thickness distribution along the beam depth.

The classic beam results have the structure shown in table 3.1 and the thickness distribution of the results are organized as table 3.2 shows.

• Resultant stresses	• Reactions	• Deformations
◦ Axial	◦ Force	◦ Displacements
. Nx	. X-react	. X-displ
◦ Moment	. Y-react	. Y-displ
. M	◦ Moment	◦ Rotation
. M-theta	. Z-moment	. Z-rot
. M-phi	◦ Psi-function	◦ Zigzag
◦ Shear	. Psi-react	. Psi-X
. Q		
. Q-theta		
. Q-phi		

Table 3.1: Classical beam results structure

• Resultant stresses	• Stresses	• Deformations
◦ Axial	◦ Sigma	◦ Displacements
. Nx	◦ Tau	. X-displ
◦ Moment		. Y-displ
. M		◦ Rotation
. M-theta		. Z-rot
. M-phi		◦ Zigzag
◦ Shear		. X-displ
. Q		
. Q-theta		
. Q-phi		

Table 3.2: 2D results structure

Chapter 4

Examples

4.1 Study of shear locking for the LRZ beam element

In this section the performance of the LRZ beam element for the analysis of a cantilever of length L under an end point load F is studied. The beam is formed by a symmetric three-layered section whose properties are described on 4.1. The analysis is made under different span-to-thickness ratios: $\lambda = 5, 10, 50, 250$ ($\lambda = L/h$).

Composite material properties			
	Layer 1 (bottom)	Layer 2 (core)	Layer 3 (top)
h [mm]	6.6667	6.6667	6.6667
E [MPa]	2.19E ⁵	2.19E ³	2.19E ⁵
ν	0.2500	0.2500	0.2500

Table 4.1: Symmetric 3-layered cantilever beam. Material properties for shear locking study

For the two first span-to-thickness ratios is made a convergence study using several meshes from one to 100 elements. Figure 4.1 shows the vertical deflection versus the number of elements. The exact stiffness matrices integration induces the shear locking phenomenon, requiring a higher number of elements, while the reduced integration of all the stiffness matrices can eliminate this undesired effect.

From figure 4.1 is chosen the S , $S\psi$ and ψ reduced integration results as reference. Figure 4.2 shows the ratio between the end node deflection obtained with different

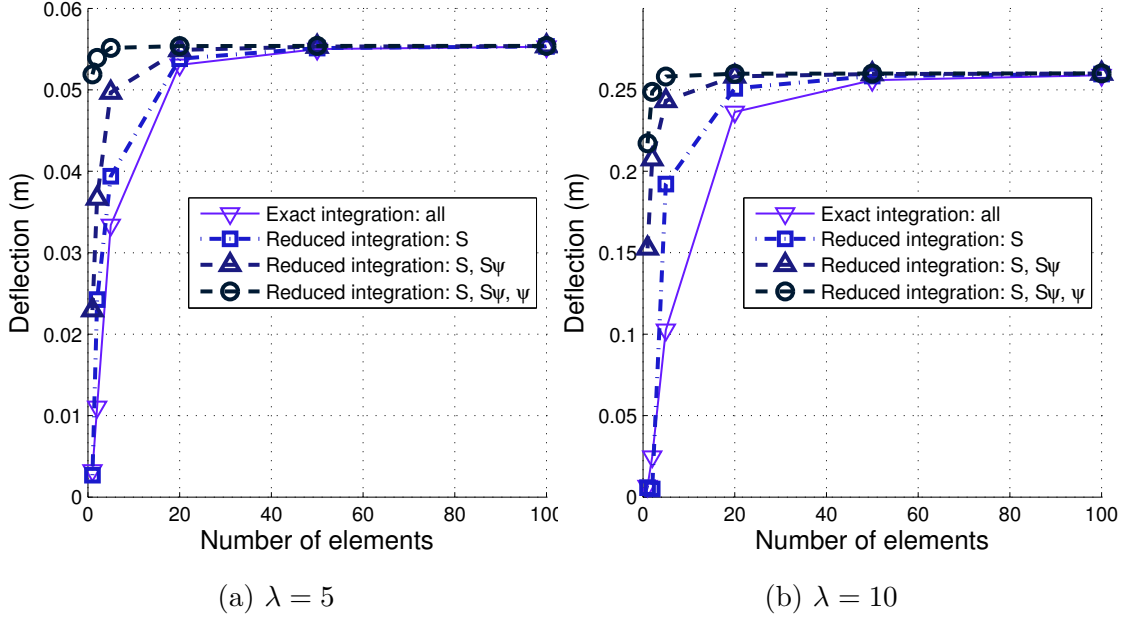


Figure 4.1: Shear locking of LRZ beam element. Convergence under different span-to-thickness ratio. Labels “all”, S , $S\psi$ and ψ refers to the matrices $\mathbf{K}_t^{(e)}$, $\mathbf{K}_s^{(e)}$, $\mathbf{K}_{s\psi}^{(e)}$ and $\mathbf{K}_\psi^{(e)}$, respectively

integration modes and the S , $S\psi$ and ψ reduced integration.

For small values of λ the reduced or exact integration of the matrix $\mathbf{K}_t^{(e)}$ leads to similar results. For slender beams, however, reduced integration is prescriptive. There is no special reason to choose $\mathbf{K}_s^{(e)}$ and $\mathbf{K}_{s\psi}^{(e)}$ or $\mathbf{K}_s^{(e)}$, $\mathbf{K}_{s\psi}^{(e)}$ and $\mathbf{K}_\psi^{(e)}$ reduced integration even though they lead to different shear stresses distribution.

More accurate studies [4] recommend using the reduced integration for matrices $\mathbf{K}_s^{(e)}$ and $\mathbf{K}_{s\psi}^{(e)}$, while matrix $\mathbf{K}_\psi^{(e)}$ should be integrated with a 2-point quadrature. In next examples will be used only reduced integration for matrices $\mathbf{K}_s^{(e)}$ and $\mathbf{K}_{s\psi}^{(e)}$.

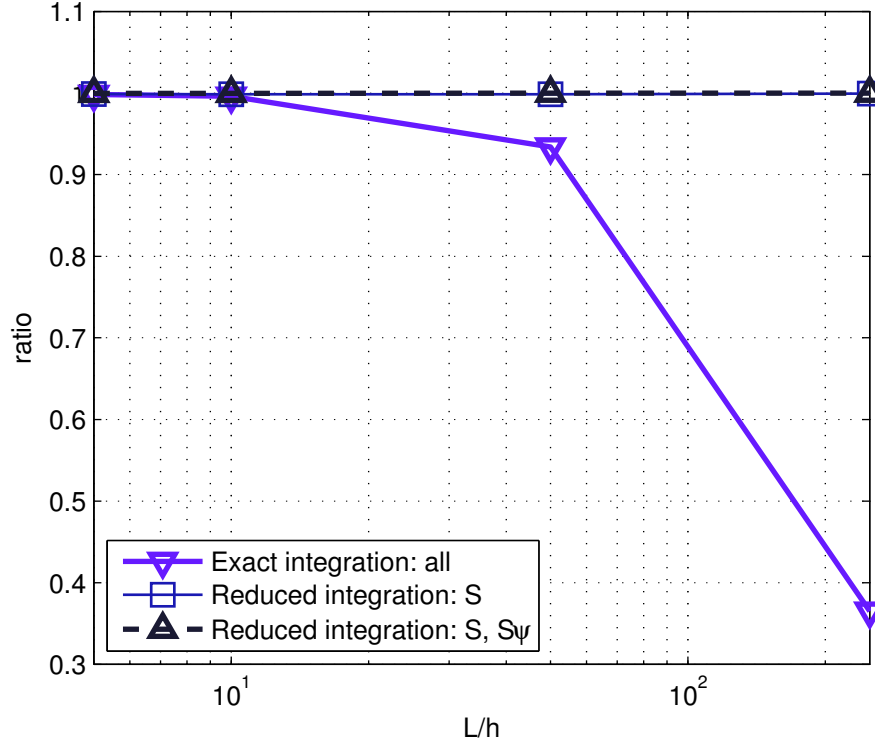


Figure 4.2: Ratio $\left(r = \frac{\omega_i}{\omega_{s,s\psi,\psi}}\right)$ versus L/h ($\lambda = 5, 10, 50$ and 250) for cantilever under point load analyzed with 100 LRZ elements

4.2 Convergence study

A simple supported beam under a uniformly distributed load of unit value is analyzed with two different composite laminated sections. Its span-to-thickness ratio is $\lambda = 10$. The section properties are listed in table 4.2. Note that the first laminate does not possess material symmetry with respect to the mid-depth reference axis, while the second composite is symmetric and holds more uniform properties.

The legend caption PS denotes the *reference* results obtained with a structured mesh of 14,400 four-noded plane stress quadrilateral elements (4.3). TBT label means the solution obtained with a mesh of 300 two-noded Timoshenko beam element. LRZ-n refers to the solution obtained with the LRZ beam element using meshes of n elements. Additionally, shear stresses can be computed from the constitutive equation (2.70) which are labeled as Nz.

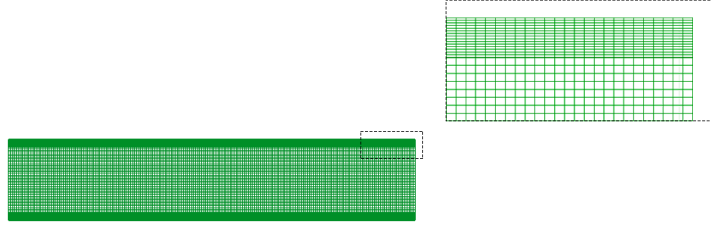


Figure 4.3: Mesh of 14,400 four-noded plane stress quadrilateral elements to obtain the “reference” results

Composite material properties				
		Layer 1 (bottom)	Layer 2 (core)	Layer 3 (top)
A	h [mm]	2	16	2
	E [MPa]	$7.30E^5$	$7.30E^2$	$2.19E^5$
	ν	0.25	0.50	0.25
B	h [mm]	6.6667	6.6667	6.6667
	E [MPa]	$2.19E^5$	$2.19E^3$	$2.19E^5$
	ν	0.25	0.25	0.25

Table 4.2: Thickness and layer properties for convergence study in a 3-layered simple supported beam under uniformly distributed load

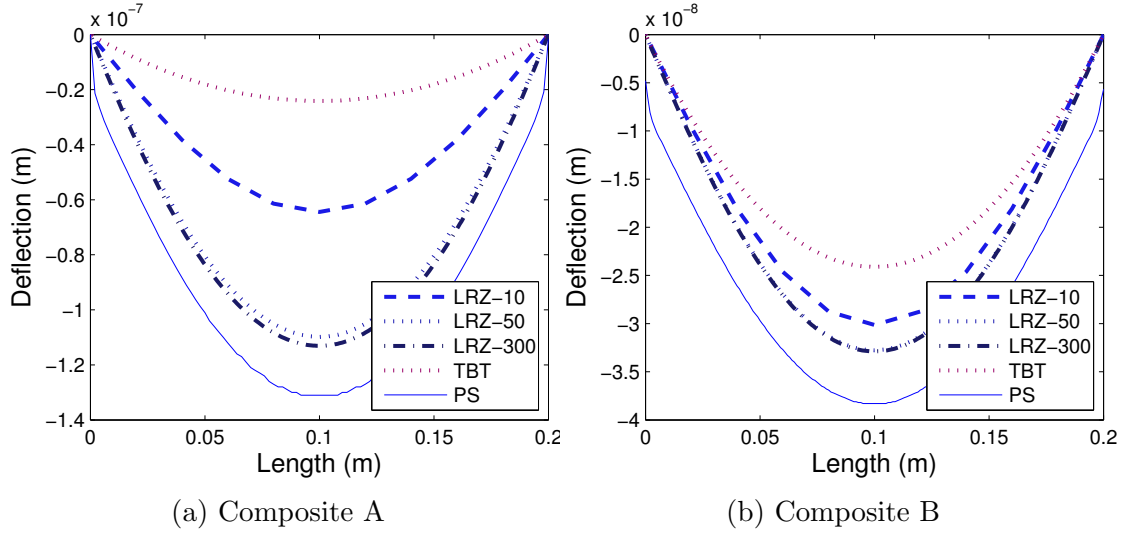


Figure 4.4: Three-layered simple supported beam under uniformly distributed load. Distribution of the deflection along beam axis

Figure 4.4 shows the deflection distribution along the beam axis for the two laminates. TBT results are considerable stiffer, the difference with the reference solution in composite A is about six times stiffer. The PS reference has some difficulties modeling the boundary conditions, note that the error is constant along the beam axis and note also, that near the fixed nodes there is an unusual deflection.

LRZ results are excellent with the 50 element mesh. Composite (A) maximum deflection relative error is less than 3% respect results obtained with the 300 element mesh. A more homogeneous case is reflected in the composite (B) which has a higher convergence, even though, the PS solution continues suffering the boundary conditions modeling.

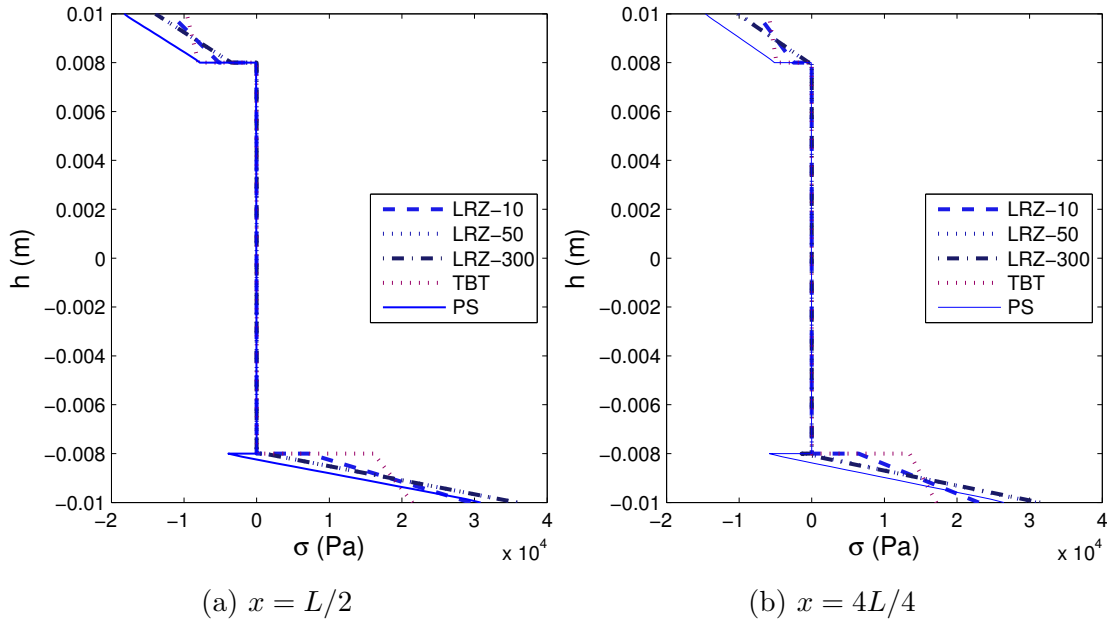


Figure 4.5: Thickness distribution of normal stresses (composite A)

The axial stresses are plot in figures 4.5 and 4.6. LRZ results agree quite well with those of the reference solution, and predict the discontinuity of σ_x better than TBT results. TBT underestimate the maximum axial stress σ_x .

Convergence is still lower for the more heterogeneous case. In composite (A) the LRZ results are displaced respect the PS reference solution. This discrepancy is due to the difference in the way the simple support conditions are modeled and the limitations of the beam theory.

Figures 4.7 and 4.8 show the shear stresses distribution along the beam depth. LRZ provides an accurate average estimate of the transverses shear stress value

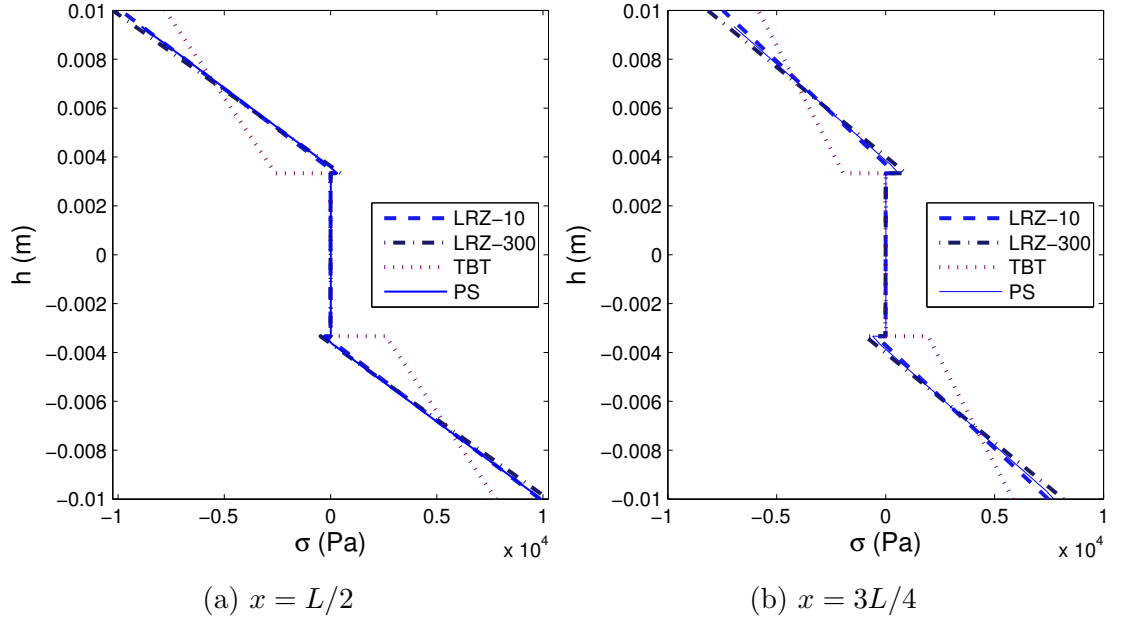


Figure 4.6: Thickness distribution of normal stresses (composite B)

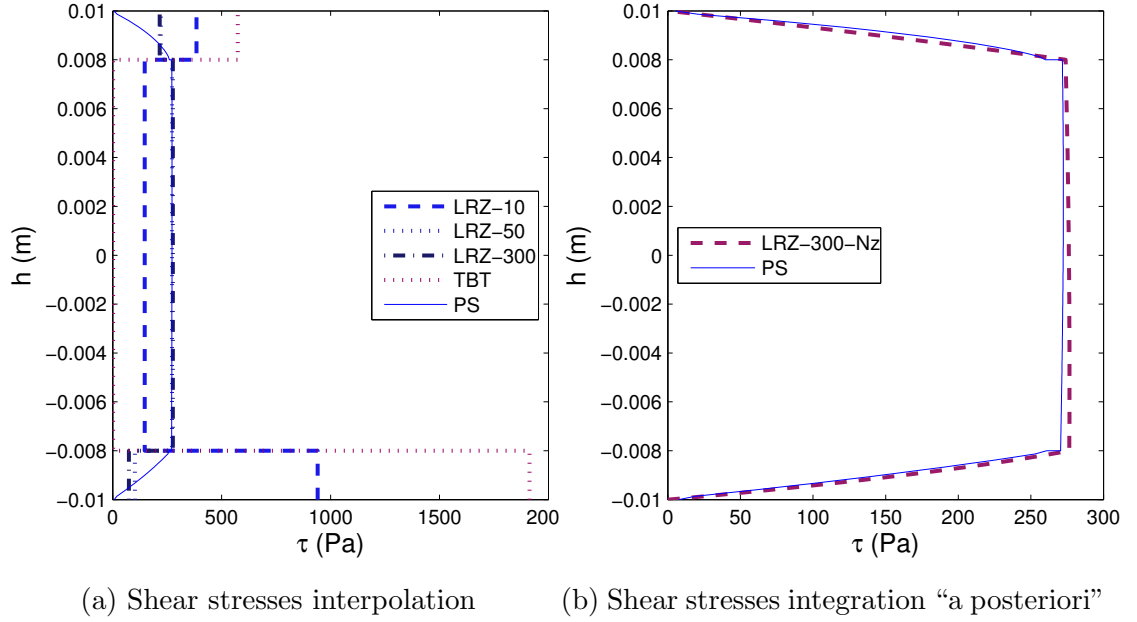


Figure 4.7: Thickness distribution of shear stresses (composite A) at $x = 3L/4$

for each layer. TBT results are clearly inaccurate. Shear stress τ_{xz} is highly overestimated by TBT at the skin layers, specially for the more heterogeneous composite (A).

The distribution of τ_{xz} across the thickness are substantially improved (figures 4.7b and 4.8b) by the integration of the equilibrium equations to compute τ_{xz} “a posteriori” as explained in section 2.2.8. This distributions of the shear stresses match the PS results.

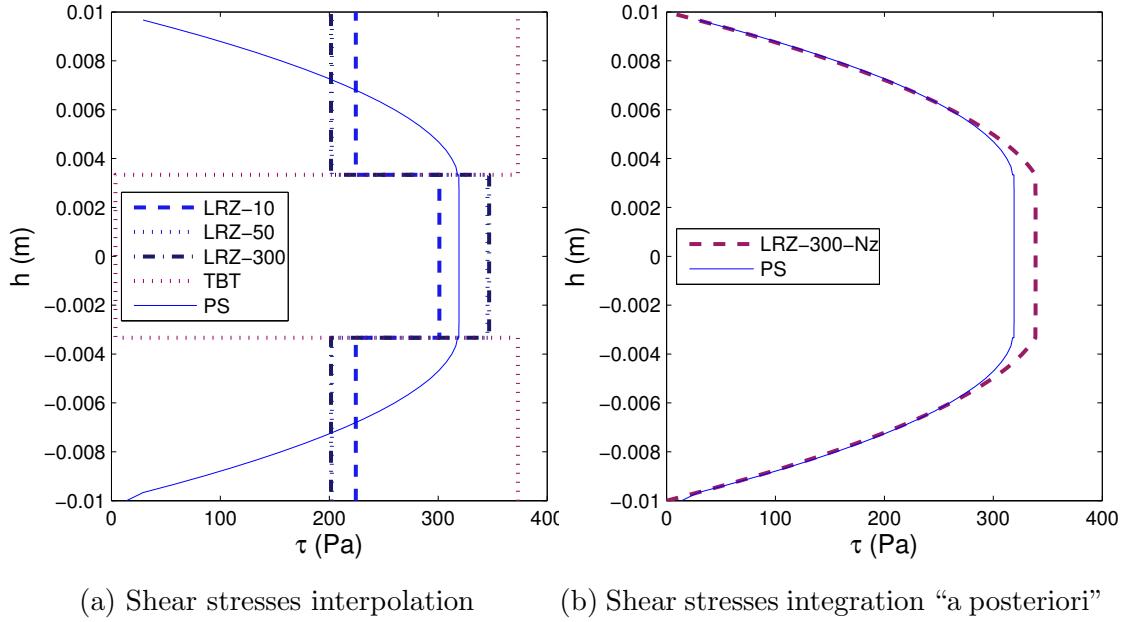


Figure 4.8: Thickness distribution of shear stresses (composite B) at $x = 3L/4$

4.3 Modeling delamination

Prediction of delamination in a composite laminated beams is a challenge for beam models. Some models introduce a kinematic unknown for each layer even though, the LRZ element can reproduce delamination without introducing additional kinematic variables. The delamination model introduces a very thin “interface” layer between adjacent materials. Delamination is produced when the material properties of the interface layer are reduced to almost zero value in comparison with those of the adjacent layers.

An example to show the capabilities of the LRZ beam element is presented. This example is a cantilever beam under end point load with a span-to-thickness ratio about $\lambda = 5$. The beam section has three layers whose properties are shown in table 4.3. A very thin interface layer has been introduced between upper and core layers (figure 4.9). The initial properties of the interface layer coincides with the those of the core layer. Next, Young modulus, and shear modulus, has been reduced up to 10 orders of magnitude (table 4.4).

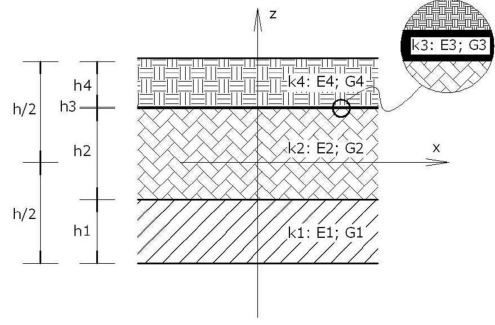


Figure 4.9: Interface layer for modeling delamination effects

Composite material properties				
	Layer 1 (bottom)	Layer 2 (core)	Layer 3 (interface)	Layer 4 (top)
h [mm]	2	16	0.01	2
E [MPa]	$7.30E^5$	$7.30E^2$	E_3	$2.19E^5$
ν	0.25	0.50	0.25	0.25

Table 4.3: Thickness and layer properties for dealmination study in a 3-layered cantilever beam under end point load. Young modulus values and it's corresponding shear modulus are given in table 4.4

The reduction of the Young and shear modulus has been applied over the whole beam length. In other cases this reduction can be applied in selected regions of the beam.

Model	E_3	G	Model	E_3	G
1	$7.38E^{+2}$	$2.43E^{+2}$	6	$7.38E^{-3}$	$2.43E^{-3}$
2	$7.38E^{+1}$	$2.43E^{+1}$	7	$7.38E^{-4}$	$2.43E^{-4}$
3	$7.38E^{+0}$	$2.43E^{+0}$	8	$7.38E^{-5}$	$2.43E^{-5}$
4	$7.38E^{-1}$	$2.43E^{-1}$	9	$7.38E^{-6}$	$2.43E^{-6}$
5	$7.38E^{-2}$	$2.43E^{-2}$	10	$7.38E^{-7}$	$2.43E^{-7}$

Table 4.4: Young and shear modulus for the interface layer for delamination study in a 3-layered cantilever beam. Values in MPa

The deflection evolution with the models is shown in figure 4.10. Note that the deflection does not change after the shear modulus of the interface layer is reduced about six orders of magnitude. Results agree reasonably well with the plane stress

model. The main difference is the additional deflection of the skin from the core layer which the plane stress model can reproduce.

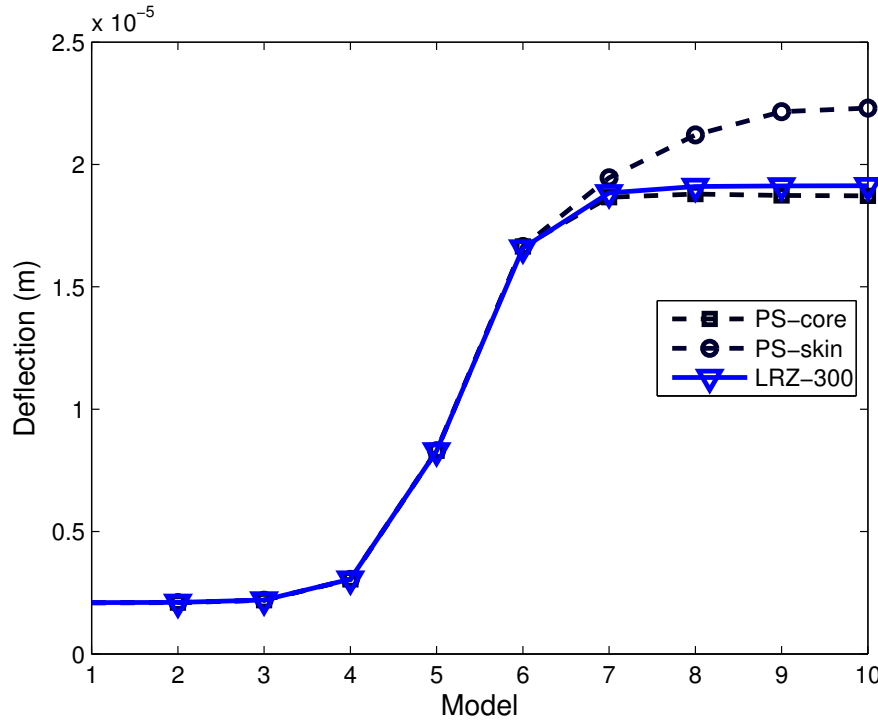


Figure 4.10: Delamination study in a cantilever beam. Evolution of end deflection with the interface layer stiffness

Figure 4.11 show the axial displacement at $x = 4L/5$ for two different models: model 2 is before delamination and model 8 is after delamination. The discontinuity of the axial displacement between the core layer and the skin layer is well represented (figure 4.11b). Results agree with the plane stress solution.

Figure 4.12 shows the thickness distribution of the axial stresses for the same models. The delamiated section works as two independent beams, there are two neutral axis. Results agree well with the plane stress ones, like in the other cases.

The thickness distribution of the shear stresses is shown in figure 4.13 for the same two models. The solution is obtained integrating the stresses equilibrium equation (equation 2.70). Note the accuracy of the solution versus the plane stress results.

This example shows the capability of the LRZ beam element predicting delamina-

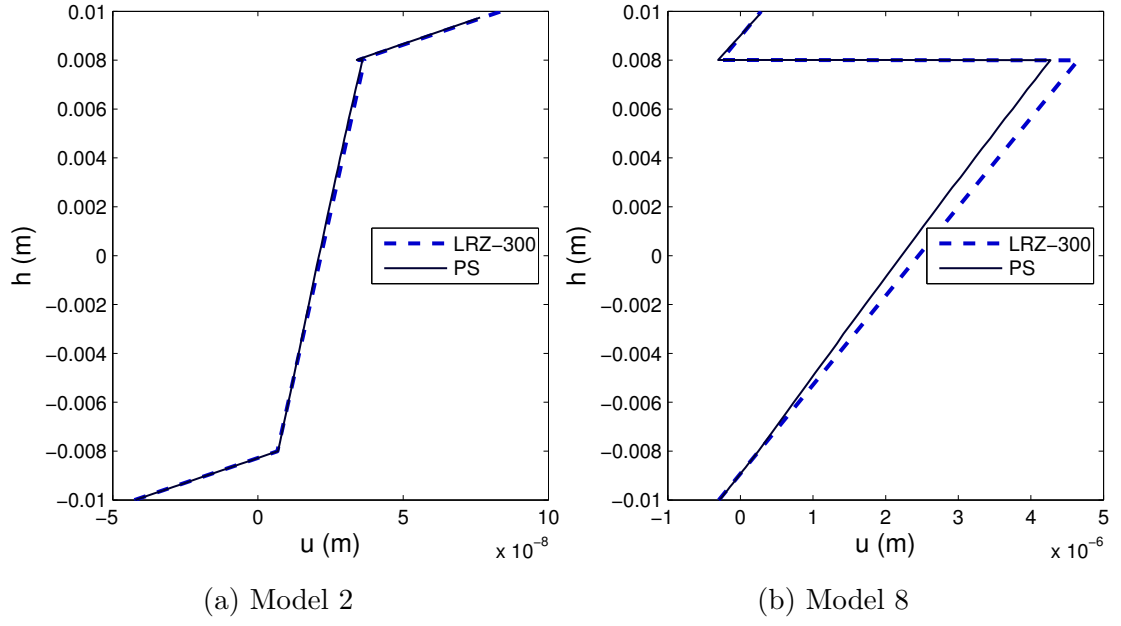


Figure 4.11: Delamination study in a cantilever beam. Axial displacements at $x = 4L/5$

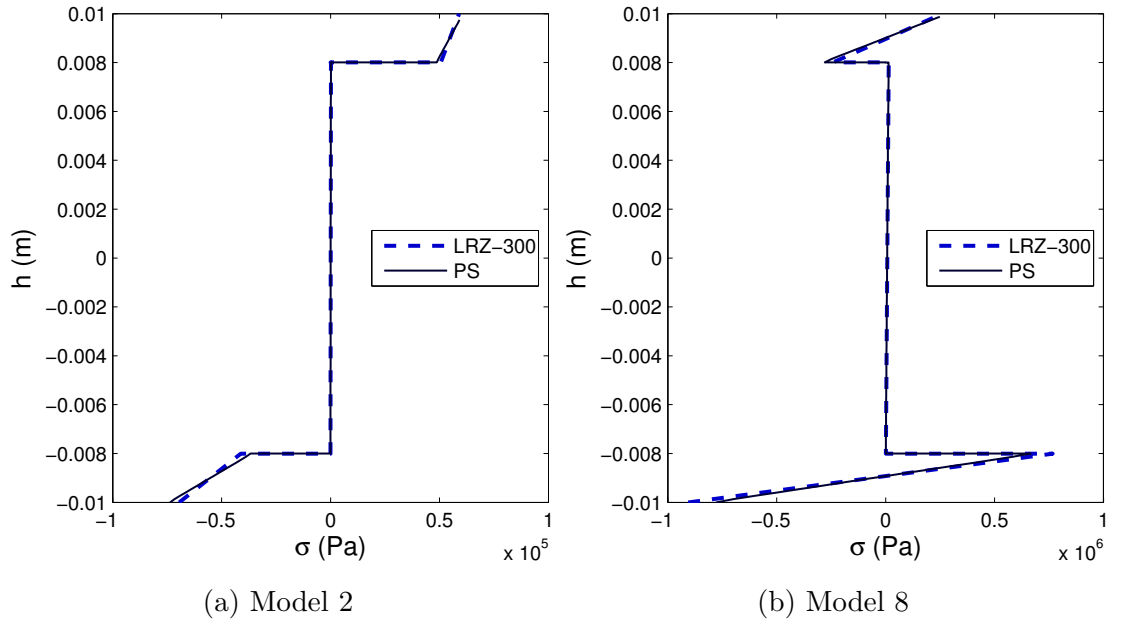


Figure 4.12: Delamination study in a cantilever beam. Thickness distribution of normal stresses at $x = 4L/5$

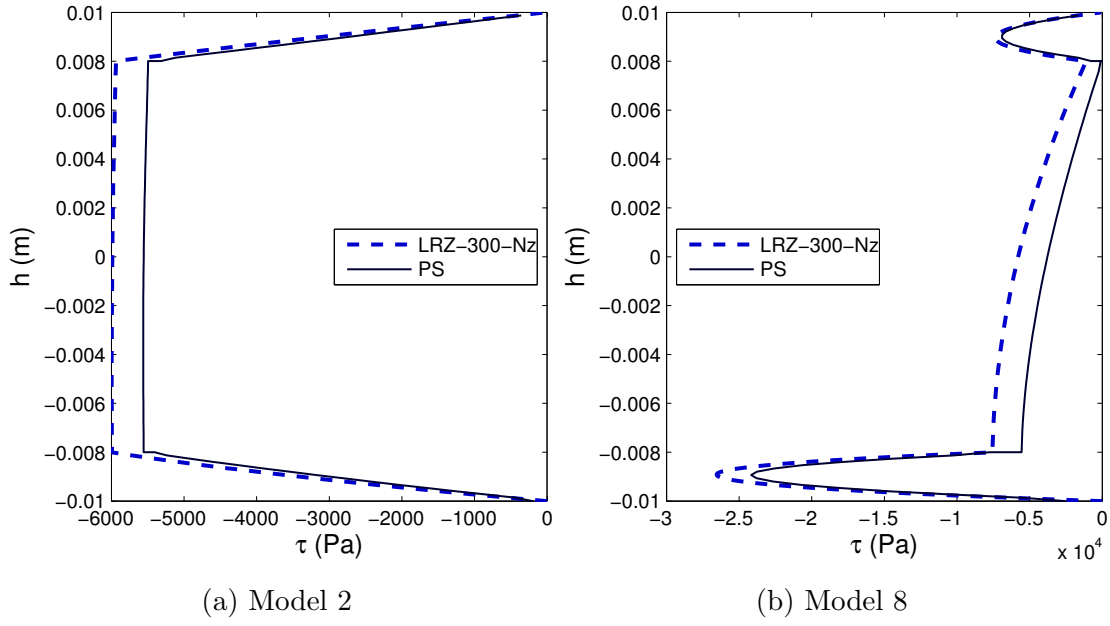


Figure 4.13: Delamination study in a cantilever beam. Thickness distribution of shear stresses at $x = 4L/5$

tion effects, a complex phenomenon for beams theories. Finally, is important to recall that the LRZ does not incorporate additional kinematic variables.

Miguel Masó Sotomayor

Chapter 5

Conclusions

This work presents two simple 2-noded beam elements. The first one is based on composite laminated Timoshenko beam theory and the second one on the refined zigzag theory.

On a regular basis the Euler-Bernoulli formulation works with one degree of freedom per node (the deflection) while the Timoshenko theory adds an additional degree of freedom per node (the deflection and the rotation). This two beam theories presented add two additional degrees of freedom per node (the axial displacement and the amplitude of the zigzag function). A standard interpolation C^0 is used for all the variables. From the tested examples we infer the following conclusions:

- The resulting linear refined zigzag element, called LRZ, is free of shear locking. It is performed by the reduced integration of the matrices $\mathbf{K}_s^{(e)}$ and $\mathbf{K}_{s\psi}^{(e)}$, while $\mathbf{K}_\psi^{(e)}$ matrix should be integrated with a two point quadrature.
- For slender isotropic beams, the bending terms dominate over the shear terms but, in composite laminated beams, the influence of transverse shear deformation can be increased drastically. This justifies the use of the advanced beam theories for composite laminated beams.
- LRZ beam element degrees of freedom does not depend on the number of analysis layers.
- The LRZ beam element numerical results agree in practically all cases of the two-dimensional plane-stress FEM results. The plane-stress FEM uses a far larger number of degrees of freedom.
- The possibilities of the LRZ beam element for predicting delamination effects has been presented in a simple but representative example.

Miguel Masó Sotomayor

Bibliography

- [1] Zienkiewicz, O.C., Taylor, R.L. y Zhu, J.Z., *El Método de los Elementos Finitos*, Vol. I Las bases, 6^a Ed., Bugeda, G., Cervera, M. y Oñate, E. (Eds.), Centro Int. de Met. Num. en Ingeniería, Barcelona, 2005
- [2] Oñate, E., *Cálculo de estructuras por el método de los elementos finitos*, 2^a Ed., Centro Int. de Met. Num. en Ingeniería, Barcelona, septiembre de 1995
- [3] Oñate, E., *Analysis with the Finite Element Method. Linear Statics*. Vol. 2 Beams, plates and shells. Centro Int. de Met. Num. en Ingeniería, Barcelona 2013
- [4] Oñate, E., Eijo, A., Oller, S., *Simple and accurate two-noded beam element for composite laminated beams using a refined zigzag theory*, Comput. Methods in Appl. Mech. Engrg., Vol. 213–216, 1 March 2012, Pages 362-382
- [5] Centro Internacional de Métodos Numéricos en Ingeniería. www.cimne.com
- [6] MAT-fem manual. www.cimne.com/projects/mat-fem, 2007
- [7] GiD customization manual. www.gidhome.com, 2012
- [8] Welch, B., *Practical programming in Tcl and Tk*, 3rd Ed., Prentice Hall PTR, 2000
- [9] MATLAB. www.mathworks.com, 2003